Extralink

MANUEL d'utilisation

V-3.4

Edité le 1er juillet 2010

Table des Matières

1. Installation 2
1.1. Définitions 2
1.2. PHASE 1 - Installation physique 2
1.3. PHASE 2 - Installation des programmes et librairies 5
1.4. PHASE 3 – utilisation des bases 5
1.5. PHASE 4 - Test de la carte de base 18
1.6. PHASE 5 - Installation des modules 19
1.7. PHASE 5 - Connexions externes 19

2. Logiciel - utilisation de la DLL XA 20
2.1. Ouverture de la connexion 21
2.2. Fermeture de la connexion 21
2.3. Sélection de l'adresse du pack 21
2.4. Sélection de l'adresse IP 21
2.5. Connexion / déconnexion 21
2.6. Détermination du time-out de réception 22
2.7. Fonctions des bases 22
2.8. Afficheur LCD 22
2.9. Fonctions des modules réadressables 23
2.10. Librairie C avec fichier de description .h (Labview) 24
2.11. Tables des fonctions générales 25

3. Alimentations et modules E/S 26
3.1. Alimentations Extralink 26
3.2. Modules de sortie tout-ou-rien 27
3.3. Modules d'entrée tout-ou-rien 29
3.4. Modules entrées analogiques 30
3.5. Modules de sorties analogiques 33
3.6. Modules de comptage 34
3.7. Passerelle RS-232 36

4. Modules de commande de moteurs 38
4.1. Module XA-TMC pour moteur micropas à course réduite 38
4.2. Cartes pour moteurs pas à pas micropas XA-MSM et XA-M3SM 41
4.3. Modules XA-GPC et XA-298 pour moteurs pas à pas 45
4.4. Modules d'asservissement de moteurs courant continu 49
4.5. Module XA-CCM moteur CC en boucle ouverte 53

1. Installation

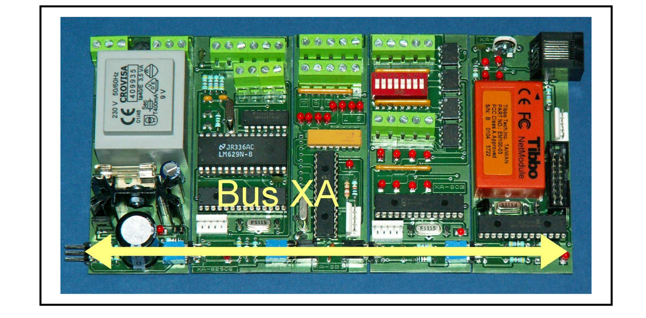
1.1. Définitions

Un **pack Extralink** est généralement constitué à partir des éléments suivants :
- une **alimentation**, qui fournit du 5 Volts et du 12 Volts régulés,
- une **carte de base**, qui assure la liaison entre l'ordinateur et les modules,
- un ou plusieurs **modules**, qui réalisent des fonctions d'interface particulières.

L'installation d'un système Extralink se décompose en 5 phases :
1 – assemblage physique
2 – installation des logiciels
3 – installation et test de la carte de base
4 – installation et paramétrage des modules
5 – connexions externes

1.2. PHASE 1 - Installation physique

Les cartes Extralink doivent être assemblées dans les rails XA-RL30 prévus à cet effet. La longueur des cartes étant de 100 mm, elles peuvent aussi être insérées dans des boitiers ou des racks pour les cartes au format « Europe ».



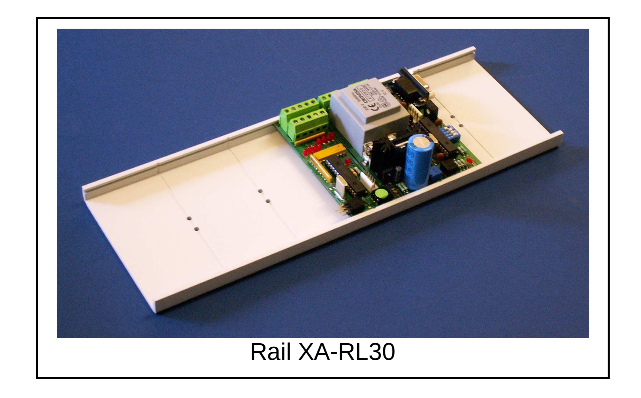
Les connexions entre les cartes se font par des connecteurs à 6 points males-femelles. Ces liaisons transportent de carte en carte les alimentations et les signaux de contrôle, et constituent le **bus Extralink**.

La carte de base, n'ayant qu'un connecteur, est toujours installée à l'extrémité droite d'un pack. L'ordre dans lequel les modules et l'alimentation sont positionnés est indifférent.

1.2.1. rails XA-RL

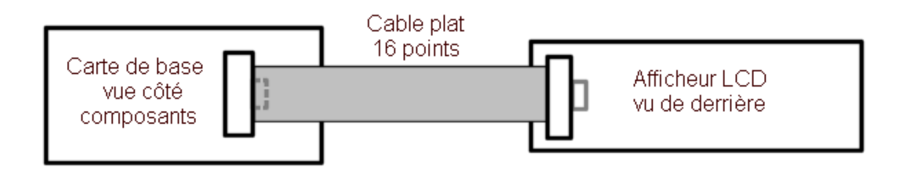
Les rails XA-RL30 ont une longueur de 30 cm et peuvent recevoir des pieds clipsables sur rail DIN.

Des marques transversales tous les 5 cm facilitent leur coupe à longueur.



Rail XA-RL30

1.2.2. Branchement des afficheurs alphanumériques

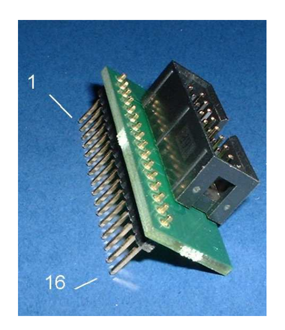


Sur toutes les cartes de base, il est possible de brancher un afficheur LCD alphanumérique (équipé d'un processeur HD 44780 ou équivalent, jusqu'à 4 lignes de 20 caractères). L'afficheur peut faciliter la mise au point des applications, mais n'est pas nécessaire au fonctionnement d'un pack Extralink.

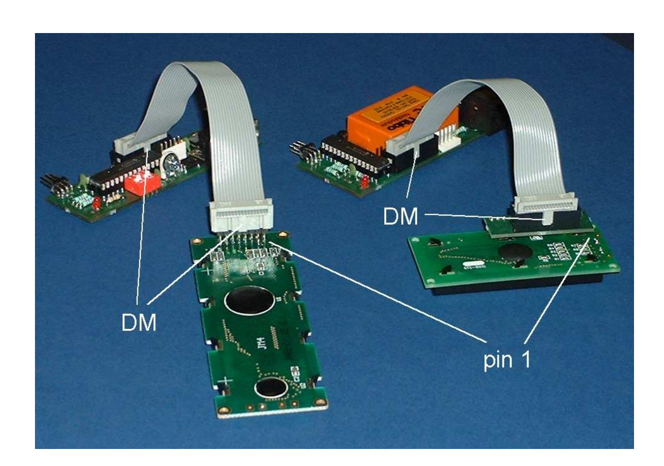
Le schéma ci-dessus indique les branchements à réaliser entre la carte de base et l'écran, à l'aide d'un câble en nappe et de 2 connecteurs sertissables HE-10 à 16 points. Les deux connecteurs doivent être d'un même côté du câble et orientés dans le même sens.

Pour faciliter le cablage des LCD équipés d'une barrette simple rangée à 14 ou 16 points, il est recommandé d'utiliser le kit de connexion XA-LCD-KIT, qui permet de reporter les connexions sur un câble en nappe. Ce kit doit être soudé par l'utilisateur au dos du LCD. Veiller à bien mettre la pin 1 du kit dans le trou 1 de l'afficheur. Sur les photos qui montrent le branchement, le détrompeur du connecteur HE10 est indiqué par « DM ».

Nous proposons un LCD simple et économique, de 1 ligne de 16 caractères référence XA-LCD1x16 équipé du kit LCD.



Kit LCD



1.2.3. Commande de l’alimentation

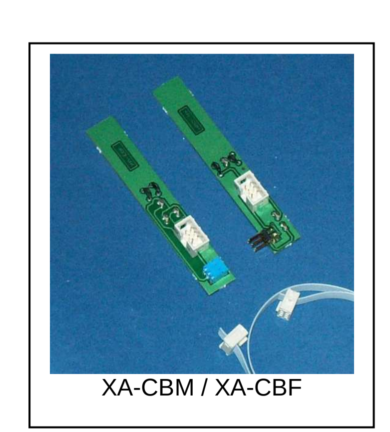
Une alimentation est nécessaire à tout pack Extralink. Dans le cas des packs XA-422, les bases reçoivent leurs tensions d’une alimentation unique.

Certaines cartes de base ont la possibilité de commander l’alimentation de leur pack, c’est-à-dire d’établir ou de couper les tensions d’alimentation sous le contrôle du logiciel. Il est ainsi possible de réinitialiser totalement le pack.

L’alimentation peut aussi être établie en permanence si on place un jumper sur la carte d’alimentation. Ce jumper inhibe la possibilité de couper l’alimentation par logiciel, et permet de garder les modules fonctionnels même en cas d’absence de l’ordinateur. En ce cas, par exemple, un module compteur pourra continuer à compter et un module de commande de moteur conservera les positions et paramètres moteur.

Les tensions fournies sont : +5 Volts et + 12Volts régulés. Si on dispose déja de ces tensions, on peut les connecter au pack XA à l’aide du bornier d’une carte XA-CBM ou XA-CBF.

1.2.4. Accessoires : prolongation du bus Extralink



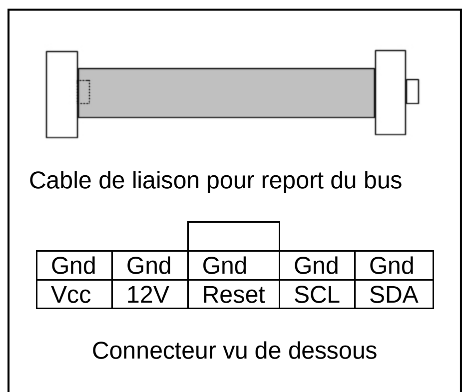
XA-CBM / XA-CBF

Les cartes XA-BF et XA-CM permettent de prolonger physiquement le bus Extralink par un câble en nappe souple. Ceci est utile par exemple si souhaite répartir les modules d’un pack entre deux boitiers.

Les connecteurs du cable plat doivent être sertis selon les figures ci-dessous : d’un même côté du cable et les deux détrompeurs dans la même direction. La longueur du câble ne doit pas excéder 50 cm.

La carte XA-CBF porte un connecteur Extralink femelle ; elle doit donc etre installée à gauche des modules. La carte XA-CBM porte un connecteur Extralink mâle et doit donc être installée à droite.

Il est possible de relier une carte CBM à une carte CBF aussi bien que CBF à CBF ou bien CBM à CBM.



Cable de liaison pour report du bus

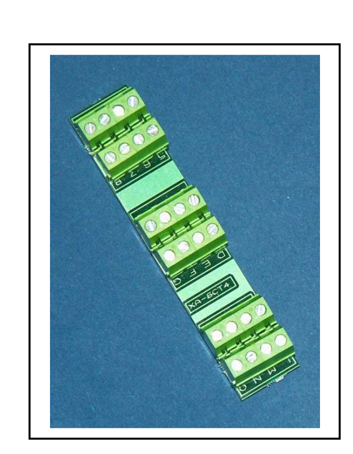
Gnd	Gnd	Gnd	Gnd	Gnd
Vcc	12V	Reset	SCL	SDA

Connecteur vu de dessous

<u>Bornier XA-CBM / XA-CBF</u>

1	2	3	4	5	6
ground	+5V	+12 V	SDA	RST	SCL

1.2.5. Accessoires : Cartes borniers



La carte XA-6CT4 peut faciliter le câblage des modules avec les équipements extérieurs. Elle comporte 6 borniers à 4 points.

Les bornes indiqués par le tableau ci-contre sont reliées entre elles.

1 – 2 – 3 – 4
5 – 6 – 7 – 8
9 – A – B – C
D – E – F - G
H – I
J - K
L – M
N - O

1.3. PHASE 2 - Installation des programmes et librairies

Tous les logiciels sont disponibles en téléchargement à partir de notre site www.sidena.com

Créez un nouveau répertoire et téléchargez :
- deux librairies : FTD2XX.DLL et XA_DLL.DLL
- le programme de test : Extralink.exe

Téléchargez également l'exemple dans le langage que vous utiliserez.

Copier les deux librairies « FTD2XX.DLL et XA_DLL.DLL dans le répertoire « C:\Windows\System\ » et/ou dans « C:\Windows\System32\ ».

Le programme Extralink.exe permet de tester complètement tous les types de bases et de modules. Ce programme peut demander la dll « QTINF70.DLL », téléchargeable sur notre site.

Pour les packs Usb XA-USB, il faut télécharger aussi le programme « Usb FTDI drivers ».
Pour les packs Ethernet XA-LAN, il faut télécharger le aussi programme « TIBBO.EXE ».
Enfin, pour les pacs XA-Digi, téléchargez le programme « Digi Discover.exe» et « Realport.Zip ».

1.4. PHASE 3 – utilisation des bases

1.4.1. Base XA-232 et XA-232-OP

Particularités

La carte de base **XA-232OP** isole la voie série et le pack par optocouplage. La carte prend l'alimentation des optocoupleurs sur les signaux modem de l'ordinateur RTS et DTR. Ces signaux doivent donc être connectés et alimentés. Ils sont gérés par le logiciel XA_DLL, qui établit leurs tensions lorsqu'on ouvre la communication.

On utilisera la carte de base **XA-232** si on ne dispose pas de ces signaux. Cette carte n'est pas optocouplée ; elle utilise seulement les signaux RX et TX.

Switch	1	2	3	4
1200	Non Utilisé	-	-	-
2400		-	-	ON
4800		-	ON	-
9600		-	ON	ON
19200		ON	-	-
38400		ON	-	ON
57600		ON	ON	-
115200		ON	ON	ON

Ces cartes nécessitent une alimentation Extralink et elles commandent cette alimentation : le logiciel peut ainsi établir ou couper l'alimentation du pack en ouvrant ou en fermant le port série par la fonction «XA ('open') ».
Pour commander l'alimentation, la carte XA-232 doit avoir le signal RTS connecté.

Largeur des cartes XA-232 et XA-232OP : 40 mm.

Installation

- Si on veut que le pack soit alimenté en permanence, positionner un jumper sur la carte alimentation. Si on veut commander la mise sous tension par logiciel, ne pas positionner de jumper sur l'alimentation.
- Positionner les dip-switches pour la vitesse de transmission souhaitée.
- Eventuellement, brancher l'afficheur LCD.
- Relier à l'ordinateur par un câble SUB-D rallonge mâle-femelle non croisé (pin 1 à pin1, pin 2 à pin 2, etc).
- Passer à la PHASE 4 : Test de la carte de base.

1.4.2. Bases XA-LAN

Particularités

La carte XA-LAN permet de connecter un pack à un réseau Ethernet.

La carte XA-LAN est construite à partir d'un module TIBBO (www.tibbo.com) . Elle se raccorde à une prise du réseau par un cordon standard RJ45 droit, ou, si elles sont raccordées directement à la prise réseau d'un ordinateur, par un cordon RJ45 croisé.

La carte alimentation doit porter un jumper pour que le pack soit alimenté en permanence : sa mise sous tension ne peut pas être commandée par logiciel.

Largeur de la carte : 40 mm.

Les deux modes de fonctionnement

Les cartes reseau peuvent fonctionner selon deux modes :
1 – le mode TCP, où l'accès au pack se fait directement par un socket TCP.
2 – le mode « Virtual Port » où un logiciel installé avec la carte ouvre un port série virtuel. La communication se fait alors comme s'il s'agissait d'un pack RS-232 dont la vitesse serait de 38400 Bps.

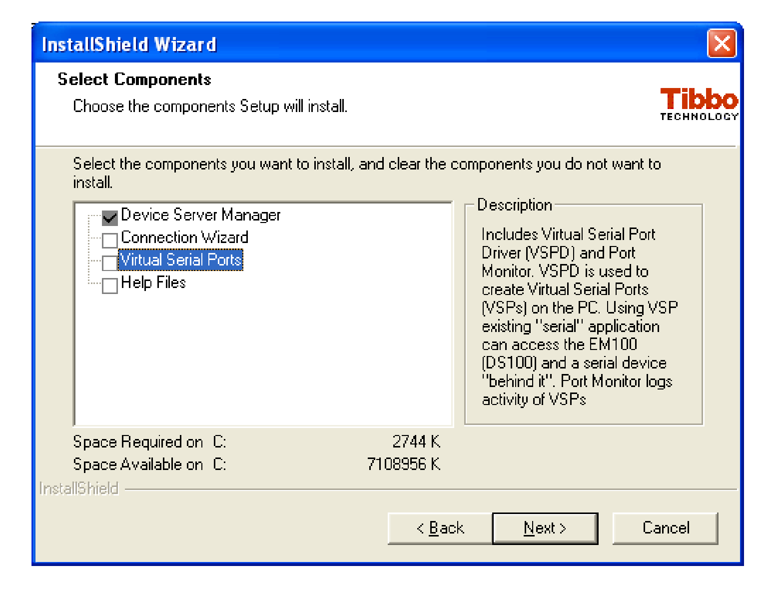
Nous donnons seulement ici les indications nécessaires pour le fonctionnement en mode TCP. La configuration du mode « Virtual Port » est décrite sur le site du fabricant du module.

Installation de la carte XA-LAN

Branchements

- Brancher uniquement la carte XA-LAN et son alimentation.
- Eventuellement, brancher l'afficheur LCD.
- Brancher au réseau.
- Mettre sous tension. La led de la carte de base doit clignoter 2 fois rapidement, puis rester éteinte.

L'afficheur doit indiquer : « Vx.x-1- » où x.x est le numéro de version du logiciel interne. Les 4 leds près du connecteur réseau doivent s'allumer brièvement, puis 2 leds restent allumées.



Logiciels XA-LAN

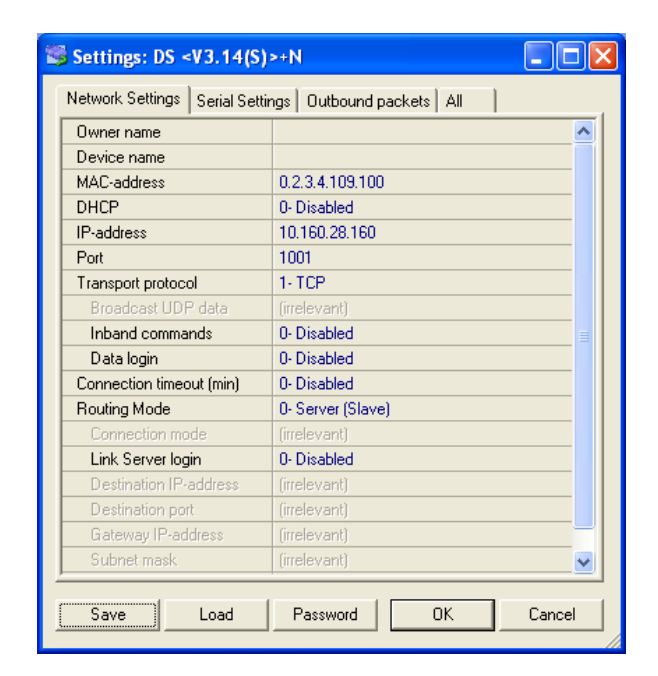
- Téléchargez le programme « TIBBO.EXE » à partir de notre site Web www.sidena.com à la rubrique «Téléchargements ».

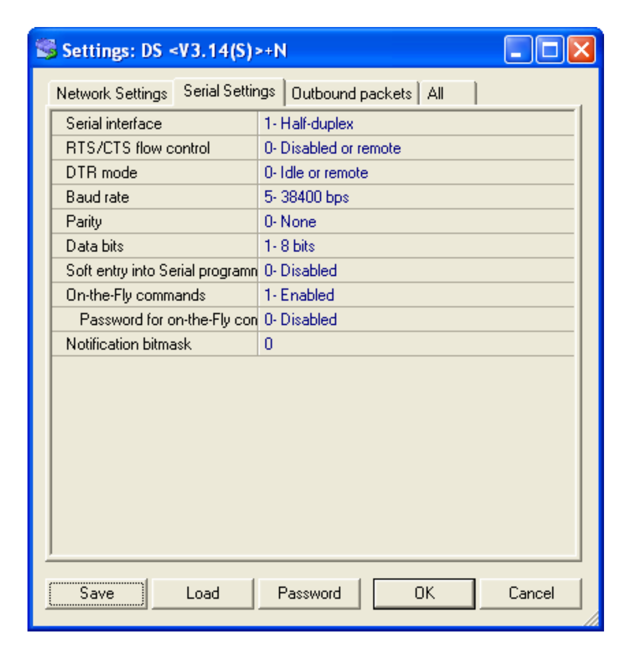
- Cliquer sur « TIBBO.EXE » pour installer le logiciel. Prendre toutes les options par défaut.
- Lancer « Démarrer programmes / TIBBO / DS Manager ».
Le programme doit trouver le contact avec le pack et il indique l'adresse IP de la carte de base.

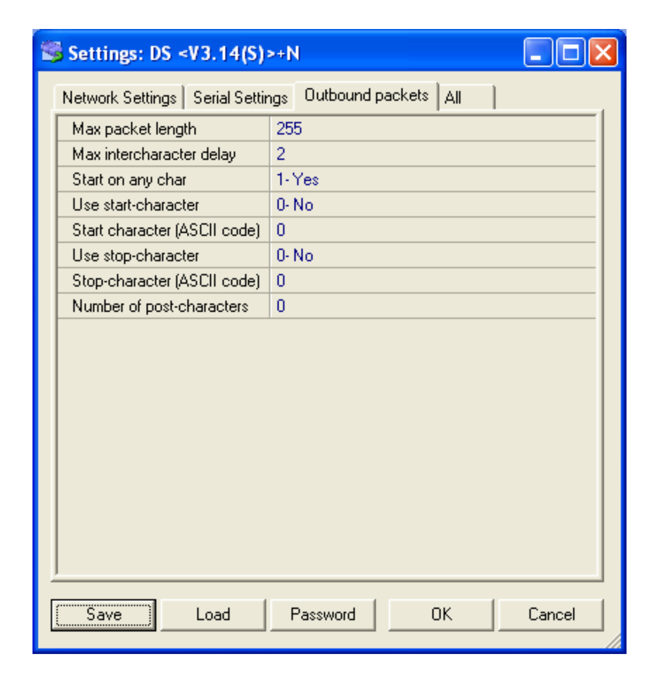
Si un message indique qu'il ne peut pas y accéder, voir plus loin « Problèmes d'accès aux packs réseau ».

- Cliquer sur la ligne correspondant au pack dans la liste pour la sélectionner
- cliquer sur « Settings ». Les propriétés doivent apparaître.

Fixer les propriétés exactement comme sur les deux écrans suivants (sauf l'adresse IP, qui sera celle que vous venez de choisir) :

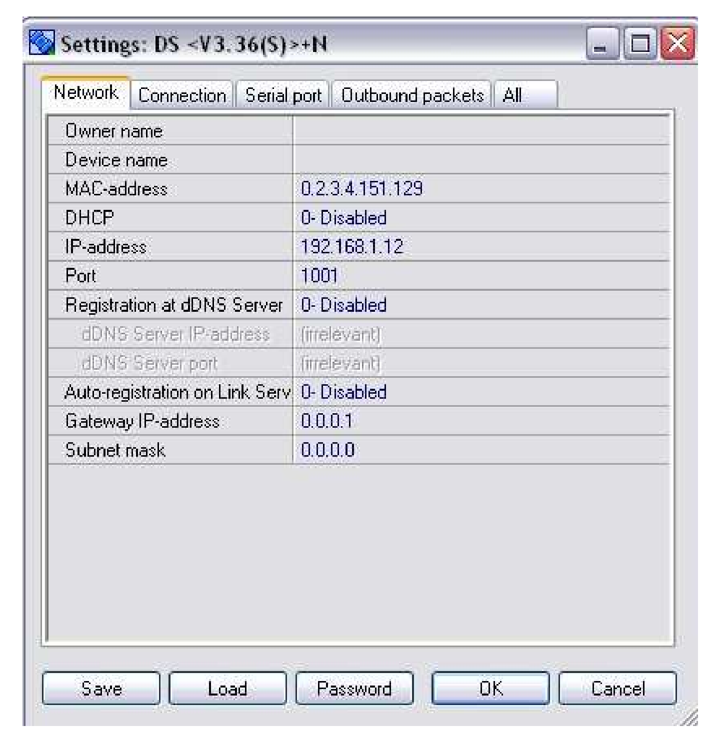


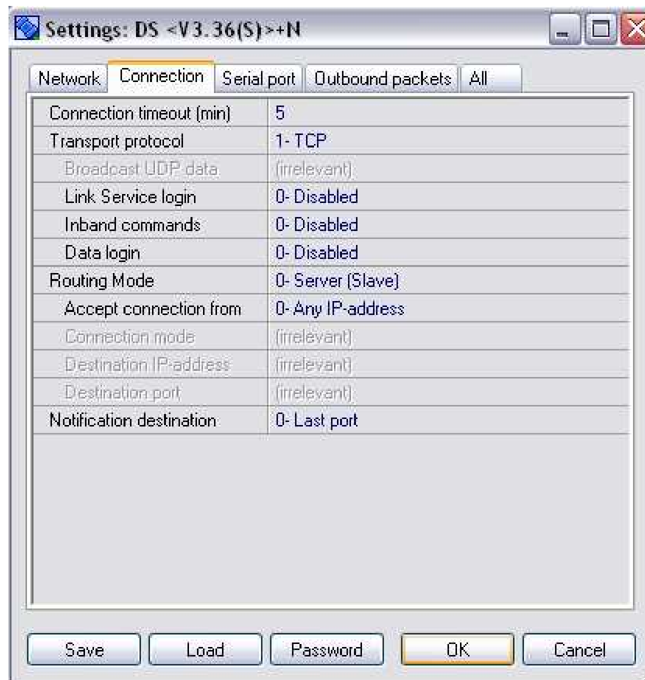


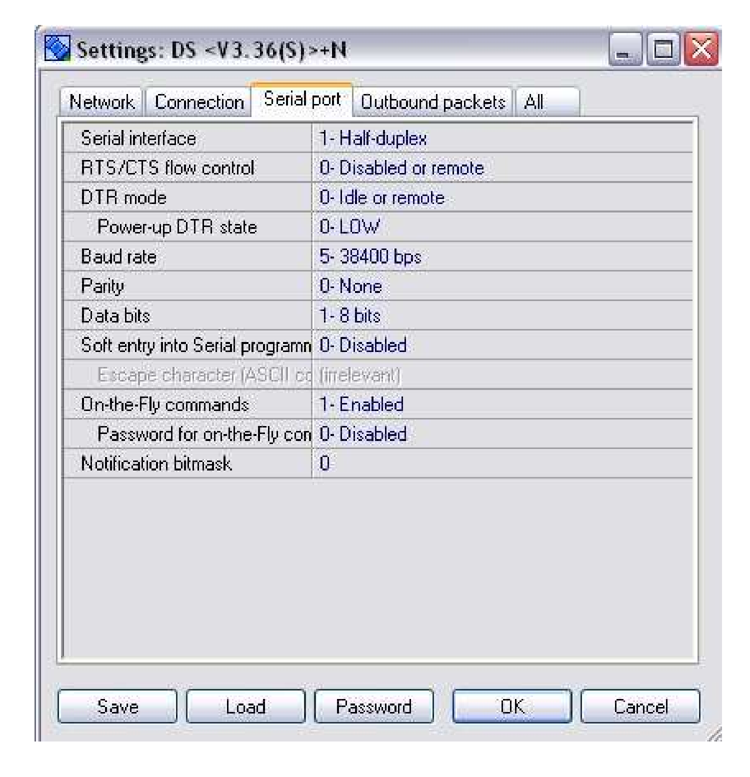


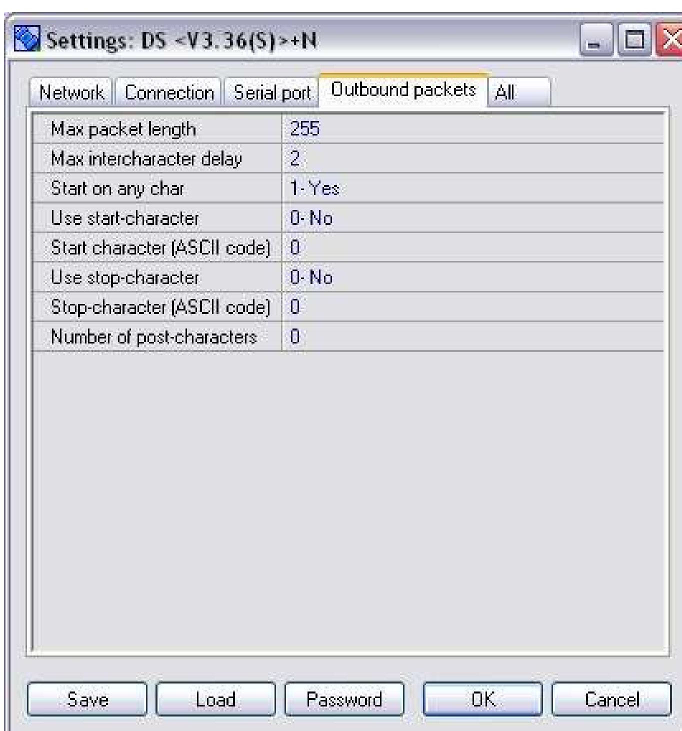
Fermez le programme. Le pack est prêt à etre utilisé.

Avec la dernière version des logiciels TIBBO, les écrans sont légèrement différents :









Problèmes d’accès aux packs réseau

Le logiciel de recherche (DS Manager) procède par recherche du numéro MAC. Il peut ne pas y accéder si un pare-feu ou un logiciel de sécurité empêche l’accès à son adresse.

En ce cas,

1 - désactivez le pare-feu Windows (sous XP : panneau de configuration > centre de sécurité)

2 – et/ou mettez « affecter une adresse IP automatiquement » dans les propiétés TCP/IP du réseau.

Le logiciel de test Extralink.exe accède par l’adresse IP. Si le logiciel de recherche trouve le pack, mais que le logiciel Extralink ne peut pas y accéder, il suffit généralement de changer l’adresse IP du pack pour la mettre dans une zone accessible :

- Lire l’adresse IP de l’ordinateur par :
  « Connexions / Connexion au réseau local / Protocole Internet TCP-IP / Propriétés»
- On lit alors une adresse telle que : 192.168.0.1 suivi du masque de sous-réseau : 255.255.255.0. Noter l’adresse IP.
- Retourner dans le logiciel de rcherche, cliquer sur « Change IP » et donner une adresse proche, par exemple : 192.168.0.2.
- Les trois premiers chiffres de l’adresse IP doivent être identiques à ceux de l’adresse IP de l’ordinateur.
- Cliquer sur « Refresh ». L’adresse IP doit apparaître dans la liste.

1.4.3. Base XA-USB

Particularités

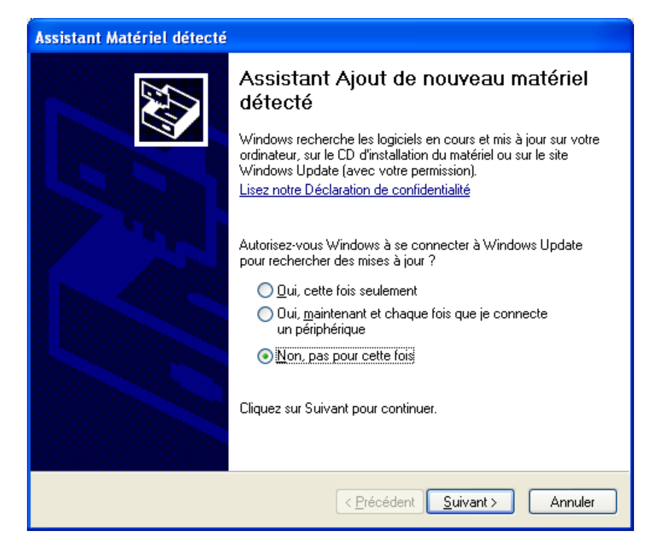
La carte de base elle-même et le LCD sont toujours alimentée par la prise USB.
Les modules sont alimentés par la carte alimentation et non par la prise USB. Ils seront alimentés en permanence si le jumper est mis sur la carte alimentation et ne seront alimentés que si la prise USB est sous tension dans le cas contraire.
Ne pas mettre de jumper sur la carte USB elle-même.
Attention : sur certaines versions de la carte, les instructions LEDON et LEDOFF sont inversées.
Largeur de la carte : 30 mm.

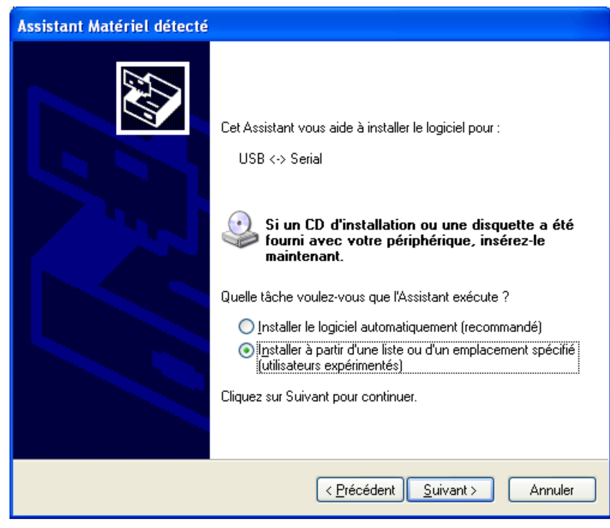
Installation

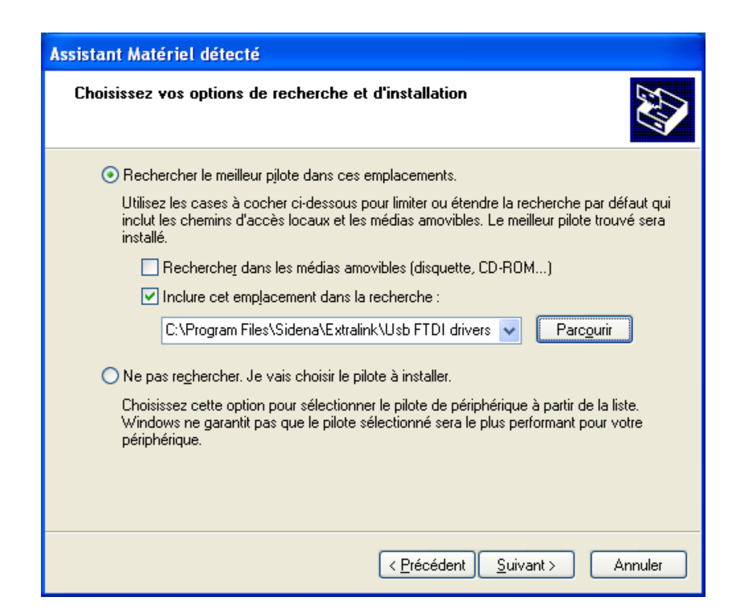
- Téléchargez le package d'installation USB à partir de notre site Web www.sidena.com à la rubrique « téléchargements ».
- Si vous avez déjà des drivers FTDI installés, il faut les désinstaller en cliquant sur « FTD2XXUN.EXE », puis sur « FTDIUNIN.EXE ». Dans le doute, il n'y a pas d'inconvénient à lancer ces deux programmes.
La carte ne doit pas être branchée pendant l'exécution de ces programmes.
- Eventuellement, brancher l'afficheur LCD.
- Connecter uniquement la carte de base à une prise USB par un câble standard type AB. La carte étant alimentée par le bus USB, il n'est pas nécessaire de connecter l'alimentation à ce stade.
- La led de la carte de base doit clignoter 2 fois rapidement, puis rester éteinte.
- L'afficheur doit indiquer : « Vx.x » où x.x indique le numéro de version.
- Le système détecte un nouveau matériel et demande ses drivers.
- Choisir le type d'installation à partir d'une liste d'emplacements spécifiés et non pas l'installation automatique.
- Indiquez le répertoire où vous avez placé les des drivers Extralink.
- Le fichier de description du Driver est FTD2XX.INF. Confirmez et attendez la fin de l'installation.
- Passer à : PHASE 4 - Test de la carte de base

La carte XA-USB est basée sur le module FT245BM de FTDI Technologies. Toutes les informations techniques sont téléchargeables sur le sîte : www.FTDI.com

Séquence des écrans d'installation :

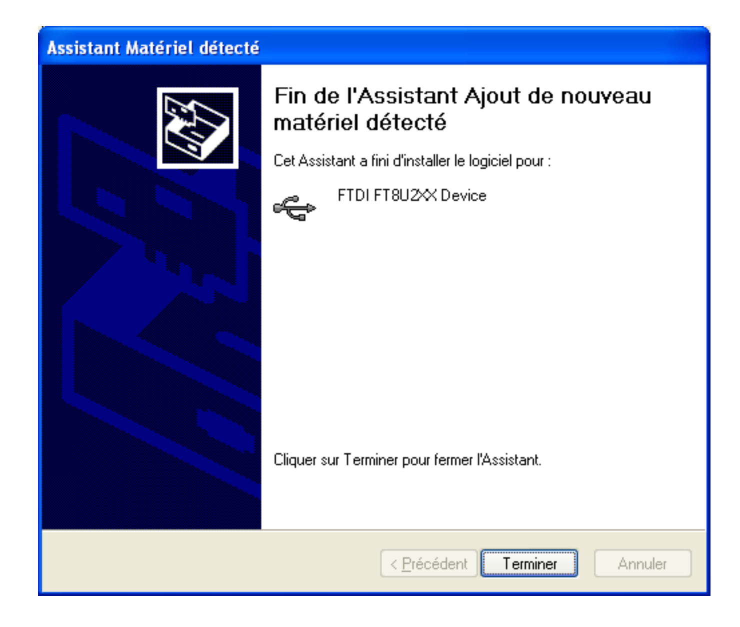






L’installation du logiciel de la carte XA-USB est terminée.

Passer à : PHASE 4 - Test de la carte de base



1.4.4. Base XA-422

Particularités

Pour pouvoir brancher les bases XA-422 il faut utiliser la carte convertisseur **XA-CNV422.** Cette carte est le point de départ du bus RS-422 qui relie tous les packs XA-422 à l'ordinateur. Les bases XA-422 sont alimentées par le bus RS-422 et ne nécessitent donc pas d'autre alimentation que celle qui est connectée à la carte convertisseur. La carte CNV-422 doit donc être branchée sur une alimentation Extralink d'une puissance suffisante pour alimenter tous les packs qui seront connectés sur le bus 422.

La carte convertisseur peut établir ou couper l'alimentation générale du bus RS-422 sous contrôle du logiciel si on ne place pas de jumper sur la carte alimentation et si le switch 1 de la carte convertisseur est sur OFF.

On peut placer jusqu'à 16 pack Extralink sur le bus.

Largeur des cartes : XA-CNV422 : 40 mm, XA-422 : 30 mm.

Paramétrage du convertisseur

- Installer le convertisseur XA-CONV422 et son alimentation.
- Choisir la vitesse de transmission sur le convertisseur.
- Le switch 1 permet de commander la tension sur le bus RS-485. S'il est sur ON, l'alimentation est connectée en permanence au bus, sinon, les packs ne sont alimentés que lorsque le programme ouvrira la communication, ce qui permet de contrôler par logiciel la mise sous tension du réseau RS-422. Ce switch a le même effet que d'installer un jumper sur la carte alimentation.
- Mettre sous tension. La led de la carte convertisseur doit clignoter rapidement, puis rester éteinte. Le nombre de clignotements de la led (de 1 à 8) correspond à la vitesse choisie.
- Mettre hors tension. Relier à l'ordinateur par un câble rallonge mâle-femelle SUB-D (pin 1 à pin1, pin 2 à pin 2, etc).

Switch	1	2	3	4	Clignotements
1200	Tension sur le bus si ON	-	-	-	1
2400		-	-	ON	2
4800		-	ON	-	3
9600		-	ON	ON	4
19200		ON	-	-	5
38400		ON	-	ON	6
57600		ON	ON	-	7
115200		ON	ON	ON	8

Paramétrage de la Base XA-422

- Réaliser le bus 422 (borniers 4 points) en reliant les borniers correspondants des cartes XA-422 : borne 1 de la carte convertisseur à toutes les bornes 1 des packs, borne 2 aux bornes 2, 3 aux 3 et 4 aux 4.
- Les connexions peuvent se faire avec du câble simple si les distances n'excédent pas quelques mètres. Pour de plus grandes distances, du câble blindé est recommandé.
- Positionner les dip-switches pour fixer l'adresse du pack et la vitesse de transmission.

Borne	1	2	3	4
	Masse	Signal RS422-A	Signal RS422-B	+12V

ATTENTION : les switches **2,3,4** de la carte convertisseur XA-CNV422 doivent avoir **exactement la même configuration** que les switches **6,7,8** des cartes de base XA-422.

Switch->	1	2	3	4
0	-	-	-	-
1	-	-	-	ON
2	-	-	ON	-
3	-	-	ON	ON
4	-	ON	-	-
5	-	ON	-	ON
6…13	…	…	…	…
14	ON	ON	ON	-
15	ON	ON	ON	ON

Switch->	5	6	7	8
1200	non utilisé	-	-	-
2400		-	-	ON
4800		-	ON	-
9600		-	ON	ON
19200		ON	-	-
38400		ON	-	ON
57600		ON	ON	-
115200		ON	ON	ON

Switches de la carte XA-422

- Eventuellement, brancher l'écran LCD.
- Mettre sous tension la carte convertisseur.
- Si le switch 4 de la carte convertisseur est sur ON, ou si un jumper est installé sur la carte alimentation, la led de la carte de base XA-422 doit clignoter 2 fois rapidement, puis rester éteinte.
- L'afficheur LCD doit indiquer «Vx.x-2-yyy-aa » où x.x est le numéro de version du logiciel interne, yyyy est la vitesse et aa est l'adresse de la carte.
- Si la carte de base controle l'alimentation, cela ne se produira qu'au moment où on exécutera la fonction XA ('open').
- Passer à : PHASE 4 - Test de la carte de base.

1.4.5. Base XA-Ascii

Particularités

La carte de base XA-ASCII est recommandée lorsqu'on ne souhaite pas utiliser la librairie Extralink, par exemple si l'on veut interfacer des modules Extralink à un automate programmable. Elle permet de commander les modules Extralink en liaison série à 9600 bps, par échange de chaines ASCII.

Cette carte exécute la quasi-totalité des commandes Extraklink ; elle permet donc de commander pratiquement tous les modules Extralink (nous consulter pour confirmation sur certains modules particuliers).

Les commandes sont envoyées sous la formes de chaines ASCII. Par exemple la chaine "LEDON 20" aura pour effet d'allumer la led du module à l'adresse 20, et la commande "SETBIT 120 3" aura pour effet de mettre à 1 le bit 3 du module à l'adresse 120.

La carte répond de la meme manière, par une chaine ASCII, en envoyant son status, et selon les cas, le status du module et la ou les réponses demandées.

La vitesse est toujours de 9600 bps, sur 8 bits sans parité.

La chaine émise par l'ordinateur se termine par Cr (caractère ASCII 13) éventuellement suivi de Newline (caractère ASCII 10), ou par Newline éventuellement suivi de CR. Les arguments sont séparés par le caractère espace.

La chaine renvoyée par la carte se termine par Cr suivi de Newline. Les éléments de la réponse sont séparés par le caractère espace.

La carte XA-Ascii est une carte XA-Mega dotée d’un programme particulier.

Logiciel

Le seul logiciel associé à la carte XA-Ascii est le programme XA-Ascii, à télécharger sur notre site.
Le programme XA_Ascii.exe permet de tester la carte et montre les messages émis et reçus.
Il donne notamement un exemple pour :
- une carte de sorties
- une carte de commande de moteurs asservis XA-629
- une carte entrées analogiques XA-8UI.

Pour tester un pack XA-Ascii :

1 - assembler le pack, le mettre sous tension. La led de la carte de base doit clignoter 3 fois et restre allumée.
2 - connecter à l'ordinateur par un cable série SUB-D 9 points male-femelle non inversé (pin 1 à pin 1, 2 à 2 etc).
3 - lancer le programme XA-ASCII, choisir le port COM et l'ouvrir.
4 - tester les boutons led ON et led OFF de la carte de base : celle ci doit s'éteindre et s'allumer.
5 - fixer l'adresse du module à tester (à gauche) ; elle est par défaut égale à 127.
6 - pour les modules réadressables, on peut tester de la meme manière la led du module et on peut changer leur adresse en mettant la nouvelle adresse (à droite) et en cliquant "changer".
7 - pour le module XA-629, on peut lancer un mouvement et lire la position.
8 - pour le module XA-8UI, on peut lire la tension sur une entrée après avoir sélectionné la gamme et l'entrée.

Ne pas oublier de fixer l'adresse du module que l'on teste dans la case "Adresse module".

La carte XA-Ascii peut donner en réponse 2 codes erreur qui lui sont particuliers :
 247 : format numerique incorrect
 246 : l’adresse module doit etre 8

1.4.6. Monopack autonome XA-MEGA

Eléments nécessaires

Pour développer une application sur XA-Mega, il est recommandé d'avoir un compilateur BASCOM-AVR et un programmateur tel que le STK-200. Il existe d'autres langages et d'autres systèmes de développement, mais ils ne sont pas supportés par cette documentation.

Le compilateur peut être acheté chez Mcselec (www.mcselec.com) et le STK200 chez kanda (www.kanda.com).

La documentation du compilateur est disponible en anglais chez Mcselec et en Français sur :
http://perso.wanadoo.fr/bils-instruments

Un afficheur LCD est pratiquement indispensable pour le développement d'une application sur XA-Mega.

Branchements

1 - brancher la carte XA-mega32 et une alimentation Extralink
2 – brancher un afficheur LCD
3 – relier le dongle du STK-200 à la fiche programmation de la carte XA-Mega par un cable plat HE10.
4 – mettre sous tension.
5 – brancher le dongle du STK-200 à la prise imprimante de l'ordinateur.

La carte XA-Mega comporte également des connecteurs permettant d'accéder directement aux ports A et C ainsi qu'au port B pin pour le branchement d'un éventuel buzzer – voir schéma ci-après..

Installation de BASCOM

Téléchargez le fichier inclus « Xabaselib.bas» et les programmes exemples Extralink à partir de notre site Web. Rangez-les dans votre répertoire de travail.

Installez le compilateur BASCOM sur l'ordinateur de développement. Faites une copie des programmes exemples BASCOM donnés avec le compilateur pour avoir une sauvegarde des originaux.

Lancer BASCOM et le configurer :
- Cliquer sur « Options/compiler/chip » et choisir « m32def.dat ».
- Cliquer sur « Options/ programmer» et choisir « STK200/STK300 programmer ».
- Sélectionner le port parallèle, généralement 378 pour LPT1.

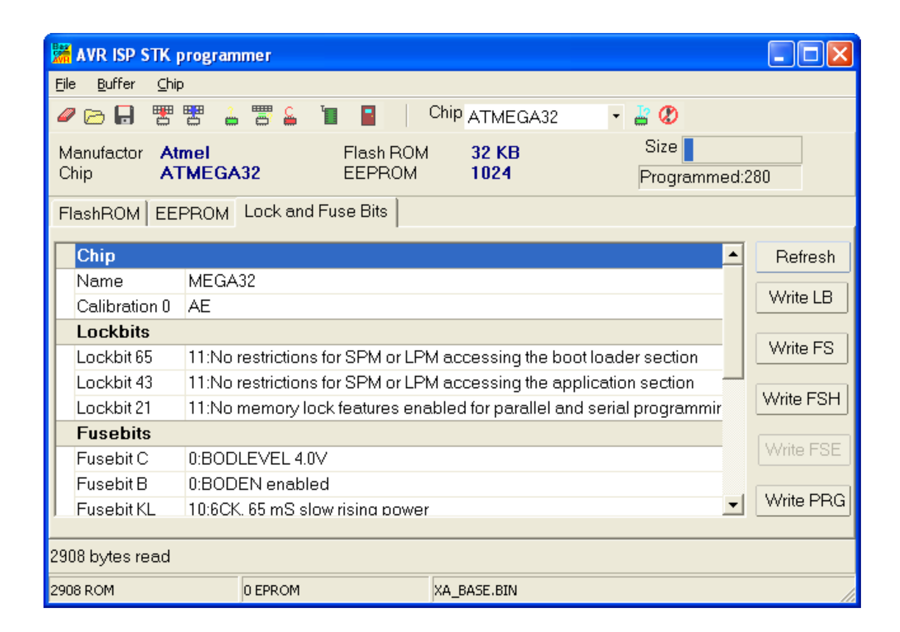
Il n'est pas nécessaire de configurer I2C, SPI et LCD si vous incluez dans vos programmes le fichier « Xabaselib.bas » ou les 5 premières lignes de ce fichier, qui sont :

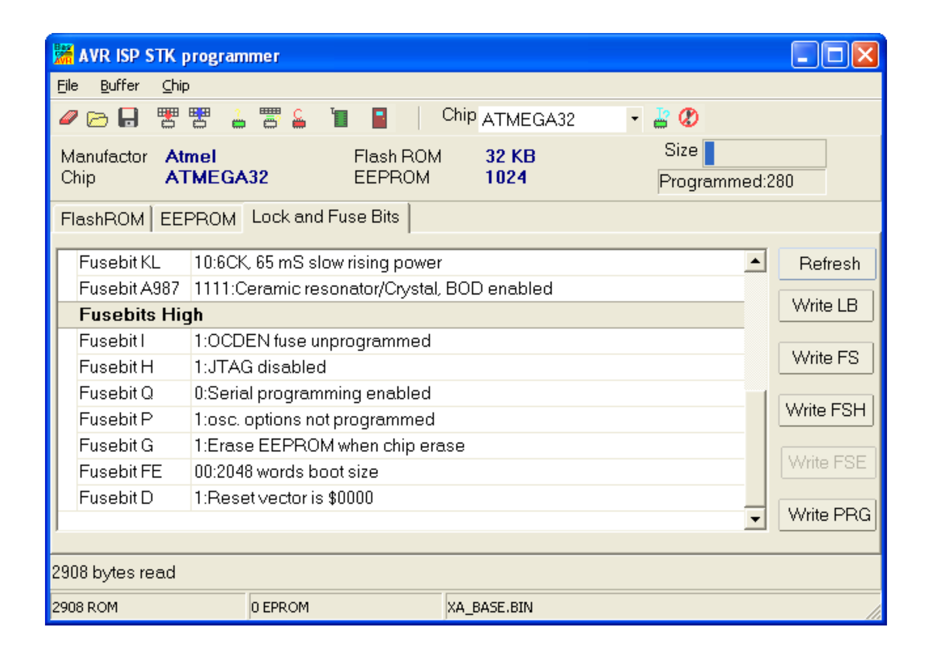
```
Config Sda = Portb.0
Config Scl = Portb.1
Config Lcdmode = Port
Config Lcdpin = Pin , Db4 = Portd.4 , Db5 = Portd.5 , Db6 = Portd.6 , Db7 = Portd.7 , E = Portd.3 , Rs = Portd.2
$crystal = 11052000
```

Programmation

Cliquer sur « chip » puis sur « identify ». Le programme doit trouver le ATMEGA32.
Charger le programme « XA_base.bas ».
Appuyez sur F7 pour compiler.
Appuyez sur F4 pour programmer le chip

Les « fuse bits » doivent être comme sur les deux écrans suivants.
Cliquer sur « chip » puis sur « autoprogram ».
La programmation dure une dizaine de secondes. Le démarrage du programme a lieu automatiquement ensuite.
La led de la carte XA-Mega doit clignoter et l'écran doit indiquer « led on / led off ».





On peut ensuite tester les modules réadressables en faisant clignoter leur led par le programme « XA_module ».

Ecriture d’un Logiciel

Le fichier inclus XA_baslelib.bas contient les définitions de configuration et les fonctions Extralink qui permettent de communiquer avec les modules.

Ces fonctions sont : XA_getb, XA_getw, XA_getl, identiques aux fonctions de la DLL, et la fonction XA définie comme suit :

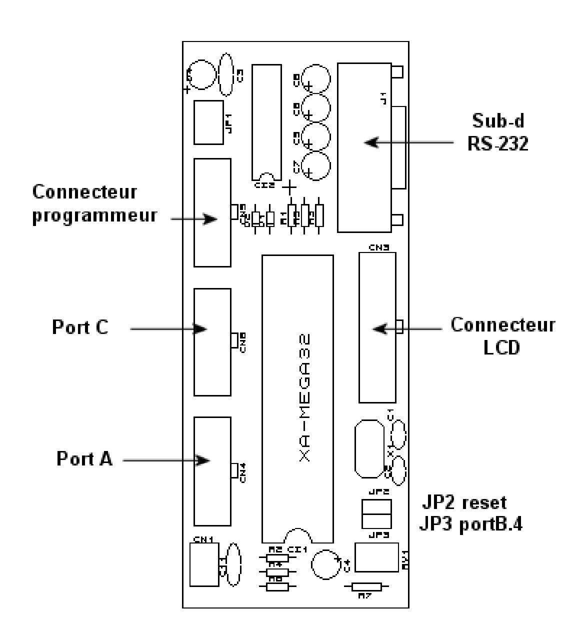
Declare Function Xa(byval Fonc As Byte , Byval Adr As Byte , Byval Arg1 As Long , Byval Arg2 As Long , Byval Arg3 As Long , Byval Arg4 As Long , Byval Arg5 As Long) As Byte.

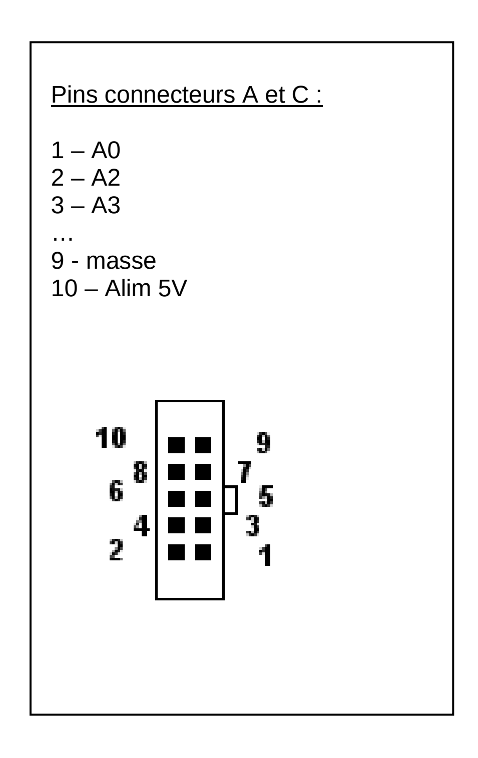
Cette fonction a toujours 7 arguments numériques :
- code fonction (byte)
- adresse I2C du module (de 0 à 127)
- arg1 à arg5 : les 5 arguments numériques

Les fonctions sont les mêmes que celles qui sont décrites dans le manuel d'utilisation Extralink, à la seule différence près qu'on doit utiliser les codes numériques et non les mnémoniques.

La correspondance est donnée dans la table suivante.

20 - WRB	60 - DELAY	120 - RDCB
21 - SETBIT	60 - WRCI	121 - RDCW
21 - WRB1	61 - BLINK	121 - RDCI
22 - RSTBIT	61 - WRCI1	123 - RDCL
22 - WRB2	62 - WRCI2	127 - RDCLL
30 - WRI	98 - WRMEM	129 - RDCWLL
31 - WRI1	99 - RDMEM	140 - SET629
32 - WRI2	100 - RDB	203 - SETADR
40 - WRL	101 - RDI	204 - LEDON
41 - WRL1	101 - RDW	205 - LEDOFF
42 - WRL2	102 - RDBI	206 - RDVER
43 - WRL3	103 - RDL	210 - I2CWR1
50 - WRCB	104 - RDBL	211 - I2CWR2
51 - WRCB1	105 - RDIL	212 - I2CWR3
52 - WRCB2	106 - RDBIL	213 - I2CWR4
53 - WRCB3	107 - RDLL	214 - I2CWR5
54 - WRCB4	108 - RDBLL	215 - I2CWR6
55 - WRCB5	109 - RDWLL	220 - I2CRD
56 - WRCB6	111 - RDLLL	252 - RDN





1.5. PHASE 4 - Test de la carte de base

Mise sous tension :

Dans un premier temps, installez seulement la carte alimentation et la carte de base. Mettre sous tension.

Si la carte alimentation porte son jumper, le pack est alimenté immédiatement et a led de la carte de base doit clignoter 2 fois rapidement, puis rester éteinte. L'afficheur doit indiquer : « Vx.x-1-yyyy » où x.x est le numéro de version du logiciel interne, et yyyy est la vitesse. Si la carte de base controle l'alimentation, ceci ne se produira qu'à l'ouverture de la communication par la fonction XA ('open').

Si on modifie la vitesse de transmission ou l'adresse fixée par les switches sur la carte de base, il faut éteindre et rallumer l'alimentation du pack pour valider la nouvelle configuration.

Ouverture de la communication :

Lancer le programme Extralink.exe.
Indiquer le type de carte de base et, selon le cas, sélectionner la voie série la vitesse de transmission.
Dans le cas d'un système XA-LAN, donner l'adresse et laisser le numéro de pack à zéro..
Cliquer sur « XA ('open') ».
Dans le cas d'un système multipack XA-422, sélectionner ensuite le pack à tester.

Test de la led et du LCD :

Faire allumer et éteindre la led de la carte de base en cliquant sur « Base led ON » et « Base led OFF ».
La zone à gauche de l'écran indique les transmissions réalisées. La réponse de la base doit être « non initialisée» au premier appel après sa mise sous tension, et « Ok » ensuite.

Si un LCD est branché, il est possible de le tester à l'aide des boutons « Clear LCD » et « LCD test », qui affiche une chaine.

Si la led de la carte de base s'allume et s'éteint correctement en cliquant sur les boutons, les transmissions entre l'ordinateur et la carte de base sont correctes. On passe alors à PHASE 4 - Installation des modules.

Cas particulier XA-LAN :

Dans le cas d'un système XA-LAN, il est possible de se connecter plusieurs packs à des adresses IP différentes. Pour cela, Changer l'adresse IP et mettre le numéro de pack à 1. Faire OPEN pour ouvrir une nouvelle connexion. Lorsque le numéro de pack est à zéro, les fonctions s'adresseront au premier pack et lorsqu'il sera à 1, elles s'adresseront au 2eme pack. Il est ainsi possible d'ouvrir jusqu'à 20 connexions.

1.6. PHASE 5 - Installation des modules

1.6.1. Principes d'adressage

Les modules sont identifiés par leur adresse I2C dans le pack. Les modules qui composents un pack doivent tous avoir une adresse différente. **Les adresse 0 à 8 et 13 à 15 sont réservées.**

Il existe deux types de modules : les modules à adresse imposée et les modules réadressables.

Les modules **réadressables** sont basés sur un microcontrolleur. Ils ont une adresse donnée à la fabrication (généralement 0 ou 127) et cette adresse peut être changée par logiciel. La nouvelle adresse est sauvegardée dans la ROM du module. Les modules réadressables ont une led qui clignote 2 fois brièvement à la mise sous tension, et que l'on peut allumer et éteindre par logiciel.

Les modules à **adresse imposée** sont construits à partir d'un composant I2C. Ils ont une adresse I2C fixée physiquement par des dip-switches ou par des jumpers.

On installe d'abord les modules réadressables, puis les modules à adresse imposée.

1.6.2. Paramétrage des adresses

- Mettre hors tension le pack et installer d'abord le premier module réadressable.
- Remettre la tension et lancer le programme Extralink.exe.
- Ouvrir la voie de communication et s'assurer que la liaison est correcte en commandant la led de la carte de base.
- Cliquer sur SCAN. Le programme indique l'adresse où le module réadressable a été trouvé. Cette adresse est généralement 127 lorsque le module n'a jamais été utilisé.
- Pour vérifier son bon fonctionnement, allumer la led du module en cliquant « Module LED ON »
- Donner une nouvelle adresse et valider. Cette nouvelle adresse doit être comprise **entre 20 et 127**. (les adresses 9 à 12 et 16 à 19 sont également disponibles, mais l'usage est de fixer les adresses des modules à partir de 20).
- Faire éteindre la led du module à sa nouvelle adresse en cliquant « Module LED OFF ».

Répéter les mêmes opérations pour les autres modules.

Le bouton « SCAN » fait apparaître la liste de tous les modules réadressables connectés. Attention : le SCAN peut durer de 1 à 2 minutes avec le pack XA-LAN et le pack XA-USB.

Installer ensuite les modules à adresse figée.

Les modules à adresse figée ne répondent pas toujours à la commande SCAN. De plus, ils peuvent interpréter SCAN comme une commande. Il faut donc éviter d'utiliser la commande SCAN lorsque des modules à adresses figée sont installés dans le pack.

1.7. PHASE 5 - Connexions externes

Mettre hors tension, relier les modules aux équipements extérieurs.

Toutes les connexions avec l'extérieur se font par des borniers à vis. Les numéros des bornes sont gravés sur le circuit devant le bornier. La borne 1 est toujours en haut à droite si on met le bus Extralink vers soi ; c'est généralement la masse ou le « moins » de la référence de tension.

La masse d'Extralink est la même que la masse du PC, sauf cas du pack XA-232OP, qui est optocouplé. Cette masse est généralement mise à la terre par la prise secteur du PC. Il ne doit y avoir aucune différence de potentiel entre cette masse et la masse des équipements reliés au système Extralink.

Se référer à la notice de chaque module pour ce qui concerne ses connexions et sa programmation.

Remettre sous tension, puis, avec le programme « Extralink.exe » tester les fonctionnalités de chaque module avec les boutons correspondant à chaque module.

2. Logiciel - utilisation de la DLL XA

Les logiciels doivent avoir été installés comme indiqué dans la phase 2. Le répertoire FTDI n'est nécessaire que pour les bases XA-USB. Les packs réseau doivent avoir été configurés.. Les deux DLL « XA_DLL » et « FT2XX.DLL » doivent être présentes dans le répertoire « Windows \ System ».

Des exemples de programmes en différents langages sont installés dans des sous-répertoires du répertoire d'installation. Ces programmes sont volontairement simplifiés pour montrer la déclaration d'une DLL et donner quelques exemples d'appels aux fonctions Extralink. Attention : ces programmes supposent une carte de base série (type 1) sur COM1 à 38400 Bps. Il vous appartient de modifier les paramètres de l'Open en fonction de votre carte de base.

Pour les systèmes de développement intégrés tels que LABVIEW, qui demandent des passages de paramètres CDECL, voir le paragraphe « Librairies C » à la fin de ce chapitre.

La librairie XA_DLL vous permet d'utiliser les fonctions **Extralink** :

XA, XA_getB, XA_getW, XA_getL, XA_str.

La principale fonction est **XA**. Elle s'écrit :

Status = **XA** (fonction : Pchar, adresse , arg1, arg2, arg3, arg4, arg5 : integer) où

- sous Delphi et VB, les arguments arg1 à arg5 sont facultatifs
- tous les arguments sont des entiers longs sur 32 bits,
- <fonction> est un pointeur sur une chaine de caractères (Pchar) indiquant la fonction demandée,
- les minuscules et majuscules sont admises indifféremment.
- <adresse> est limitée à 127.

Cette fonction retourne la valeur zéro si l'opération a réussi, sinon la valeur retournée indique le type d'erreur qui a été rencontrée. S'il s'agit d'une opération d'écriture ou de lecture dans une carte ou un module, la valeur retournée indique si la transmission a été correcte. Lorsqu'une base ou un module vient d'être mis sous tension, il retourne la valeur 1, ce qui indique qu'il a été réinitialisé. Ensuite, il doit toujours renvoyer la valeur zéro.

De la même manière, la fonction **XA_str** (fonction :Pchar, adresse : integer , chaine : Pchar) envoie une chaine de caractères.

Les messages d'erreur sont donnés en clair par la fonction **XA_errmsg** (code_erreur, 0), qui renvoie le message sous la forme d'une chaine.

Dans le cas d'une lecture, les réponses reçues sont extraites de la réponse du pack par les fonctions :
B = XA_getB si la valeur est sur un octet
W = XA_getW si la valeur est sur 2 octets
L = XA_getL si la valeur est sur 32 bits.

Ces fonctions doivent être toujours exécutées dans l'ordre indiqué par la notice.

Fonctions annexes

D'autres fonctions peuvent faciliter l'écriture des programmes :

XA_errmsg (code_erreur : byte) : Pchar	retourne un pointeur sur le message d'erreur
XA_lastgetB : integer	retourne la valeur du dernier GetB effectué
XA_coderr : integer	retourne le code erreur de la dernière fonction XA effectuée.

Exemple :

Lecture du numéro de version du logiciel d'une carte de base. Ce numéro est composé de 2 octets. Les cartes de base sont toujours à l'adresse 8.

```
If XA ('rdver', 8) > 1 then                -> envoie la fonction « RDVER » à l'adresse 8
  Afficher (XA_errmsg (XA_coderr))         -> si erreur, afficher le message en clair
Filler = XA_getB                           -> le 1er octet est toujours égal à zéro
Version = XA_getB                          -> fournit le numéro de version
Sous_version = XA_getB                     -> fournit le numéro de sous-version
```

2.1. Ouverture de la connexion

La version 2 des logiciels pemret d'ouvrir plusieurs voies de communication dans un même programme.
La voie de communication active à un moment donnée est fixée par :

Fonction XA ('ACTIVEPACK', num) où num est compris entre 0 et 99. Le numero du pack actif est initialisé à zéro, il n'est donc pas nécessaire d'utiliser cette fonction si on travaille sur un seul pack.

Fonction XA ('open', Npack, Ncom, Baudrate)
Npack est le type de pack utilisé.

Cette fonction ouvre la voie de communication (réelle ou virtuelle) correspondant au type de pack spécifié et elle établit l'alimentation sur le ou les packs connectés.

XA ('open', 0) pour XA-USB
XA ('open', 1, Ncom, Baudrate) pour XA-232
XA ('open', 2, Ncom, Baudrate) pour XA-422

- Baudrate = vitesse de transmission. Elle doit être précisé seulement pour les cartes RS232 et RS422, en fonction de la disposition des switches sur la carte de base. Les valeurs possibles sont : 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
- Ncom = numéro de la voie série (1=COM1, etc) - doit être précisé dans le cas d'une liaison série.

Pour XA-232 et XA-422, la liaison est normalement ouverte avec 2 stop bits. On peut ne spécifier qu'un seul stop bit en ajoutant le parametre 1 à la fin : **XA ('open', 1, Ncom, Baudrate, 1**) .

Pour Les packs réseau fonctionnant en mode TCP, utilise la fonction XA_str et non la fonction XA en spécifie l'adresse IP :

XA_str ('OPEN', 0, adresse_IP)

2.2. Fermeture de la connexion

Fonction XA ('close', Npack) où Npack est le type de pack utilisé.

2.3. Sélection de l'adresse du pack

Function XA ('adrpack', adresse_pack)

Cette fonction ne s'utilise qu'avec les bases XA-422 et radio XA-RF, car plusieurs packs peuvent être connectés. Elle spécifie le pack auquel seront adressés les messages suivants.
Par défaut, adresse_pack est égal à zéro.

2.4. Sélection de l'adresse IP

Function XA_str ('adrip', adresse_IP)

Cette fonction ne s'utlise qu'avec les packs en mode TCP socket. Elle spécifie l'adresse IP du pack sous une forme telle que : « 192.168.010.015' ».
Attention : c'est la fonction XA_str et non la fontion XA qui est utilisée ici.

2.5. Connexion / déconnexion

Function XA ('CONNECT, 0) et XA ('DISCONNECT, 0)

Ces fonctions ne s'utilisent qu'avec les packs réseau en mode TCP socket, après que la fonction OPEN ait été exécutée.
L'ordinateur doit se connecteur au pack pour pouvoir réaliser une transmission. Lorsque le pack est connecté, aucun autre ordinateur ne peut y accéder. Il faut donc déconnecter si on veut libérer l'accès au pack.

2.6. *Détermination du time-out de réception*

Le time-out de réception est la durée après laquelle le logiciel constate une absence de réponse de la base. Il est normalement fixé par défaut à l’ouverture selon le type de base utilisée. Cependant, il est possible de le modifier par la fonction :

Function XA (‘TIMOUT’, duree) où <durée> est en millisecondes.

Il peut être utile d’augmenter cette valeur particulièrement dans le cas de communication Ethernet sur des réseaux chargés, où les temps de réponse peuvent fluctuer .

2.7. *Fonctions des bases*

En réponse à un message, les bases renvoient un premier octet qui est toujours égal à zéro. Ceci permet d’être compatible avec les modules réadressables, car ceux-ci renvoient un premier octet qui indique leur état.

Les cartes de base comportent une led qu’il est possible d’allumer et d’éteindre.
Leur logiciel interne porte un numéro de version sur 2 octets :

- premier octet : version
- deuxième octet : sous-version.

On peut écrire un octet dans la mémoire de la base et le relire.

Pour toutes ces opérations, les ***bases sont à l’adresse 8***.

XA (‘ledon’, 8)	allumage led
XA (‘ledoff’, 8)	extinction led
XA (‘rdver’, 8)	lecture numéro de version
XA_getB	fournit le premier octet , toujours égal à zéro
XA_getB	fournit le numéro de version
XA_getB	fournit le numéro de sous-version
XA (‘wrmem’, 8, valeur)	écrit l’octet <valeur> en mémoire
XA (‘rdmem’, 8)	relit l’octet
XA_getB	fournit le premier octet , toujours égal à zéro
XA_getB	fournit l’octet mémoire

2.8. *Afficheur LCD*

Les cartes de base permettent de brancher un écran LCD alphanumérique.
Il répond aux fontions suivantes :

XA (‘lcdset’, 8, commande)	commande = 1 pour effacer l’écran
	commande = 2 retour position 1er caractère
	commande = 12 masquer curseur
	commande = 14 curseur apparent
	commande = 15 curseur clignotant
	commande = 128 curseur au début de la 1ere ligne
	commande = 192 curseur au début de la 2eme ligne
	commande = 148 curseur au début de la 3eme ligne
	commande = 212 curseur au début de la 4eme ligne

Les positions du curseur, qui correspondent aux 4 dernières commandes, peuvent varier selon les constructeurs.

XA_str (‘lcdstr’, chaine, adresse) affiche une chaine sur l’écran à la position courante

2.9. Fonctions des modules réadressables

ATTENTION : les modules réadressables (et seulement les modules réadressables) renvoient un premier octet qui est l'état du module. Lorsqu'on fait une lecture sur un module réadressable, il faut donc toujours lire cet octet par XA_getB pour avoir accès aux octets suivants de la réponse. La valeur 0 indique un état normal et les valeurs 1 ou 101 indiquent que le module vient d'être mis sous tension. Les autres valeurs indiquent une erreur, qui dépendent du type de module.

2.9.1. Changement de l'adresse d' un module

L'adresse d'un module réadressable est par défaut égale à 127. Elle doit être changée pour une nouvelle adresse, comprise entre 20 et 126, par la fonction suivante :

XA ('setadr', ancienne_adresse, nouvelle_adresse)

2.9.2. Allumage de la led – lecture de la version d'un module

Comme les bases, tous les modules réadressables comportent une led qu'il est possible d'allumer et d'éteindre.

XA ('ledon', adrmod)	allumage led
XA ('ledoff', adrmod)	extinction led

Ils ont un numéro de version de leur logiciel interne sur 2 octets :
- premier octet : version
- deuxième octet : sous-version.

XA ('rdver', adrmod)	lecture numéro de version
XA_getB	fournit le status du module
XA_getB	fournit le numéro de version
XA_getB	fournit le numéro de sous-version

Exemple : lecture du numéro de version du logiciel d'un module.

```
If XA ('rdver', 22) <> 0 then            -> exécute la transmission
  Afficher (XA_errmsg (XA_coderr)        -> si erreur, afficher le message base en clair
If XA_getB <> 0 then                     -> status du module, doit etre zéro
  Afficher (XA_errmsg (XA_lastgetB)      -> sinon afficher le message module en clair
Version = XA_getB                        -> fournit le numéro de version
Sous_version = XA_getB                   -> fournit le numéro de sous-version
```

2.10. Librairie C avec fichier de description .h (Labview)

Certains systèmes de développement, tels que **LABVIEW** ou VEE de Agilent Technologies, demandent une bibliothèque compilée compatible langage C accompagnée d'un fichier de description. Dans ce cas, il faut déclarer l'import de la librairie AX_DLL avec le fichier de description « Extralink.vh », que l'on trouve dans le répertoire « VEE Example ». On emploira ensuite les fonctions c_getB en lieu et place de getB, c_XA au lieu de XA, etc… Les fonctions c_XA sont identiques aux fonctions XA, mais utilisent le standard de passage de paramètres CDECL au lieu de STDCALL.

Dans LABVIEW, il arrive que le port soit laissé ouvert à la fin de l'exécution d'un programme. Il est donc prudent d'exécuter l'instruction « CLOSE » avant de faire l'open, close n'a aucun effet si le port est fermé.

Table des messages d'erreur

0 : OK
1 : Base not initializated
11 : Base frame error
12 : Base Overrun error
13 : Base I2C time out
14 : Base too long message
15 : Base Escape character eror
16 : Base checksum error
17 : Base message length error
18 : Base received message too short
19 : Module unknown function
101 : Not initialized
102 : Open error
103 : Comm not open
104 : Message empty
105 : No answer or truncated answer
106 : Erreur checksum from pack
107 : Too long answer
108 : Bad argument
109 : Bad character inside answer
110 : IP send error
111 : IP receive error
112 : IP close error
113 : Other send error';
114 : IP connexion error
115 : Too many submessages
116 : Too long message
240 : Unknown command from module
255 : Unknown function

2.11. Tables des fonctions générales

Les fonctions auxquelles répondent toutes les bases et tous les modules réadressables sont résumées dans les tables suivantes, où on retrouve :

- la valeur de la fonction (1 octet)
- le nombre d'arguments
- le type de réponse, avec les conventions suivantes :
    - le status envoyé par les modules réadressables est noté S.
    - les éléments notés B sont des octets, en lecture ils sont obtenus par GETB
    - les éléments notés W sont sur 16 bits, en lecture ils sont obtenus par GETW
    - les éléments notés L sont sur 32 bits, en lecture ils sont obtenus par GETL.

Fonctions des cartes de base : adresse = 8

Fonctions	adresse	arguments	Réponse	Commentaire
open	8	selon base	-	Ouvrir la connexion
close	8	B	-	Fermer la connexion
adrpack	8	B	-	Fixer l'adresse du pack
ledon	8	-	-	Allumer la led
ledoff	8	-	-	Eteindre la led
rdver	8	-	B+B	Lecture du numero de version
wrmem	8	B	B	Ecrit l'octet mémoire
rdmem	8	-	-	Lit l'octet mémoire

Fonctions générales pour tous les modules réadressables

Fonction	Adres.	arguments	Réponse	Commentaire
setadr	De 9 à 127	B	S	Changement de l'adresse I2C
ledon		-	S	Eteindre la led
ledoff		-	S	Allumer la led
rdver		-	S+B+B	Lecture du numero de version
wrmem		B	S+B	Ecrit l'octet mémoire
rdmem		-	S	Lit l'octet mémoire

3. Alimentations et modules E/S

3.1. Alimentations Extralink

3.1.1. Alimentations secteur XA-PS200 – XA-PS600

L'alimentation PS200 ne doit pas dépasser 200 mA en 5 V et 200 mA en 12volts, et 600 mA pour la PS600.
Si le jumper est positionné sur la carte, le pack sera alimenté en permanence. Sinon, le pack sera alimenté sous le controle du logiciel, à l'exécution de la fonction «XA ('open') ».
La led est le témoin d'alimentation du pack.
Le fusible est de 250 V / 300 mA rapide pour la PS200 et 600 mA pour la PS600.
La terre du secteur est connectée à la masse du pack.
Largeur : 40 mm.

Branchements

Borne	1	2	3	4	5	6
	Masse	+5 Volts	+12 Volts	Secteur 220V	Terre=masse	Secteur 220

3.1.2. Alimentation XA-12V

Cette alimentation doit être alimentée par une tension 12 Volts continus régulés telle qu'une batterie. Elle ournit une tension 5V. (Ne pas dépasser 1A sur le 5V).
Si le jumper est positionné sur la carte, le pack sera alimenté en permanence. Sinon, le pack sera alimenté sous le controle du logiciel, à l'exécution de la fonction « XA ('open') ».
La led est le témoin d'alimentation du pack.
Le fusible est de 250 V / 2 A rapide.
Largeur de la carte : 25 mm.

Branchements

Borne	1	2
	Masse	+12 V

3.1.3. Alimentation XA-ALCC

Cette carte doit être alimentée par une tension 12 à 24 Volts continus entre les bornes 1 et 2.
Si la tension d'entrée est supérieure à 14 Volts, ne pas placer le jumper J2 (près du régulateur). Si Elle est inférieure à 14 Volts, placer le jumper J2 qui court-circuite le régulateur 12 Volts. En ce cas, le 12 Volts sera la tension d'entrée, non régulée.
Ne pas dépasser 0,5A en 5V et 0,5A en 12 Volts.
Si le jumper J1 (près du connecteur Extralink) est positionné sur la carte, le pack sera alimenté en permanence. Sinon, le pack sera alimenté sous le controle du logiciel, à l'exécution de la fonction «XA ('open') ».
La led est le témoin d'alimentation du pack.
Le fusible est de 250 V / 2 A.
Largeur de la carte : 25 mm.

Branchements

Borne	1	2	3	4	5
	Masse	+Vin	Masse	+5V	+12V

3.2. Modules de sortie tout-ou-rien

Modules à 2, 4 et 8 relais Reed XA-2REED et XA-4REED
Pouvoir de coupure 300 mA - 100 Volts.
Seuls les contacts travail (normalement ouverts) sont disponibles.
Les relais REED ne sont pas recommandés pour commander des charges inductives.
Largeurs :
XA-2REED : 30 mm.
XA-4REED : 40 mm.
XA-8REED : 60 mm.

Modules à 2 et 4 relais mécaniques XA-2RM et XA-4RM
Pouvoir de coupure 10 A - 250 Volts.
Les contacts repos sont disponibles seulement sur XA-4RM.
XA-2RM : largeur 30 mm.
XA-4RM : largeur 50 mm.

Modules à 2 relais statiques XA-2RSCA et XA-2RSCC
Commutation au zéro de tension.
Existent en deux versions : CA pour commuter des courants alternatifs et CC pour des courants continus.
pouvoir de coupure en CA : 3A - 12 à 245 Vac.
pouvoir de coupure en CC : 3A - 2 à 60 Vcc.
Largeur : 30 mm

Modules à 4 relais statiques XA-4RS
Pour courant alternatif seulement.
Commutation au zéro de tension.
Pouvoir de coupure : 1,5A – 12 à 280 VAC.
Ne nécessitent pas la tension d'alimentation 12V (5V seulement).

Connexions des modules à relais

Borne	1	2	3	4	5	6	7	8
	Relais 0	Relais 0	Relais 1	Relais 1	Relais 2	Relais 2	Relais 3	Relais 3

Pour les modules XA-4RM seulement :

Borne	9	A	B	C
Au repos en contact avec borne	2	4	6	8

Modules à transistors

Caractéristiques communes :
Les sorties actives sont mises au moins. La charge doit donc être entre le plus d'une alimentation et une sortie. Chaque sortie contient une diode d'écrétage de surtension. Le moins de l'alimentation doit être connecté à la borne 1 et le plus à la borne A.

Module XA-8SD : 8 drivers ULN2803A – 500 mA / 50 Volts maxi par sortie. Ce module n'est pas isolé : le moins de l'alimentation est la masse du pack.

Borne	1	2	3	4	5	6	7	8	9	A
	- alim	Sortie 1	Sortie 2	Sortie 3	Sortie 4	Sortie 5	Sortie 6	Sortie 7	Sortie 8	+ alim

Module XA-4MOS : 4 transistors Mosfet optocouplés. Courant maxi 5A sous 50 Volts. Les sorties sont isolées du pack par optocouplage.

Borne	1	2	3	4	5	6
	- alim	Sortie 1	Sortie 2	Sortie 3	Sortie 4	+ alim

Programmation

Opérations de sortie simples :
On positionne l'ensemble des sorties par la fonciton XA ('WRB', adresse, valeur) où valeur est un octet dont chaque bit représente l'état d'une sortie.
On peut positionner individuellement chaque sortie par les fonctions 'SETBIT' et 'RSTBIT'.

Temporisations et clignotements :
La fonction DELAY permet d'activer une sortie pendant un temps donné.
La fonction BLINK permet de faire clignoter une sortie à une fréquence donnée (clignotement).
L'unité de temps pour les clignotements et les temporisations est de 1/100 de seconde.

Lectures :
La fonction RDB permet de lire l'état des sorties. Elle doit être suivie de GETB pour lire le status du module, puis de GETB.
En cas de clignotement ou de temporisations, la fonction RDW lit le temps restant avant le prochain changement d'état.
Elle doit être suivie de GETB pour le status du module, puis de GETW pour la lecture.

Fonction	Adres.	arguments	Réponse	Effet
wrb	De 9 à 127	B	S	Forcer les 8 sorties à la valeur B (0..255)
setbit		B	S	Forcer la sortie B (0..7) à 1
rstbit		B	S	Forcer la sortie B (0..7) à zéro
delay		B+W	S	Activer la sortie B (0..7) durant le temps W (16 bits)
blink		B+W	S	Clignotement de la sortie B (0..7) avec la période W (16 bits)
rdb		-	S+B	Lecture de l'état des sorties
rdcw		B	S+W	Lire le temps restant avant changement de la sortie B (0..7)

3.3. *Modules d'entrée tout-ou-rien*

Module XA-8BI
8 entrées pour contacts secs ou niveaux TTL. Chaque entrée est munie d'une pull-up et d'une led au + 5 Volts. Une entrée non connectée est à 1 et elle est à 0 si elle est mise à la masse.
Largeur : 30 mm

Borne	1	2	3	4	5	6	7	8	9	A
	masse	Entrée 0	Entrée 1	Entrée 2	Entrée 3	Entrée 4	Entrée 5	Entrée 6	Entrée 7	masse

Module XA-8OB
8 entrées optocouplées. Les entrées sont électriquement isolées du pack. Attention : chaque entrée est normalement à 1. Pour mettre une entrée à 0, il faut lui fournir un tension entre +5 et +15 Volts si le switch corespondant est sur ON, entre +10 et +26 Volts si le switch correspondant est sur OFF. Le moins de la tension d'entrée doit être connecté à l'une des bornes notées V-, c'est à dite la borne 1 ou la borne A.

Largeur : 40 mm.

Borne	1	2	3	4	5	6	7	8	9	A
	V-	Entrée 0	Entrée 1	Entrée 2	Entrée 3	Entrée 4	Entrée 5	Entrée 6	Entrée 7	V-

Opérations de lectures simples :
La fonction RDB renvoie un octet dont chaque bit représente l'état d'une des entrées.
La fonction RDCB suivie du numéro de l'entrée (de 0 à 7) renvoie 1 si l'entrée est à 1 et renvoie 0 si l'entrée est à 0.

Exemple : if XA ('RDB', adresse) <> 0 then afficher « erreur base»
If XA_getB <> 0 then afficher « erreur module »
Entrees = XA_getB

Impulsions et changements d'état :
Pour capter des évènements fugitifs tels que des détecteurs de chocs, on définit d'abord l'état normal de chacune des entrées par la fonction WRB. Par exemple : XA ('WRB', adresse, 15) indique que les 4 entrées de poids faibles doivent normalement être à 1 et les 4 entrées de poids forts doivent être à zéro.
On lit ensuite par la fonction RDW, qui renvoie 2 octets. Le premier octet est l'état des entrées au moment de la lecture, comme la fonction RDB. Le deuxième octet indique si chaque entrée a été dans l'état anormal depuis la dernière lecture. Par exemple, si l'entrée 2, qui devait rester à 1, est passée à 0, le deuxième octet sera 00000010, même si l'entrée 2 est revenue à l'état 1.
La fonction RDI doit être suivie de GETB pour le status du module, puis de deux fois GETB.

Utilisation en compteur :
Chaque entrées d'un module d'entrée peut fonctionner en mode compteur. On peut lire le temps écoulé depuis la dernière lecture, ce qui permet de connaître exactement le nombre d'impulsions par unité de temps pour des mesures de fréquence.
La fréquence d'entrée est limitée à 500 Hz.
Les fonctions WRCB, WRCB1 et WRCB2 fixent le mode de fonctionnement souhaité pour chaque entrée.
Les compteurs sont lus par les fonctions RDCL (compteur seul) ou RDCLL (compteur + temps écoulé depuis la dernière lecture) suivi du numéro de l'entrée, de 1 à 8. Ces fonctions doivent être suivies de GETB pour le status du module, puis d'une ou 2 fois GETL selon qu'on lit le temps ou non.

Fonction	**Adres.**	**Arg.**	**Réponse**	**Commentaire**
Wrb	De 9 à 127	B	-	Fixer état normal des entrées
rdb		-	S+B	Lit les entrées sous la forme d'un octet
rdw		-	S+B+B	Lit entrées et changements d'états depuis la dernière lecture (1)
Wrcb		B	S	Met l'entrée B en mode compteur sur front montant
Wrcb1		B	S	Met l'entrée B en mode compteur sur front descendant
Wrcb2		B	S	Met l'entrée B en mode compteur sur front montant et descendant
Wrcb3		B	S	Ni le compteur B ni le temps ne seront pas remis à zéro à la lecture
Wrcb4		B	S	Le compteur B sera remis à zéro à la lecture
Wrcb5		B	S	Le temps sera remis à zéro à la lecture
Wrcb6		B	S	Le compteur B et le temps seront remis à zéro à la lecture
RDCL		B	S+L	Lit le compteur B
RDCLL		B	S+L+L	Lit le compteur B et le temps écoulé

3.4. Modules entrées analogiques

3.4.1. Module XA-8UI

Ce module n'est pas réadressable. Il permet de lire 8 entrées analogiques dans les gammes de tension suivantes : 0 à 5V, 0 à 10V, -5 à 5V, -10 à +10V. Les tensions d'entrée doivent être comprises entre –16,5 et +16,5 Volts par rapport à la masse (risque de détérioration du convertisseur au-delà). La précision de la conversion est de 12 bits (1/4096).

Largeur : 30 mm

Connexions

Borne	1	2	3	4	5	6	7	8	9	A
	Masse	Entrée 0	Entrée 1	Entrée 2	Entrée 3	Entrée 4	Entrée 5	Entrée 6	Entrée 7	+5V

SW1	SW2	SW3	SW4	adresse
Non utilisé	off	off	off	40
	off	off	ON	41
	off	ON	off	42
	off	ON	ON	43
	ON	off	off	44
	ON	off	ON	45
	ON	ON	off	46
	ON	ON	ON	47

L'adresse du module peut être fixée de 40 à 47 selon la position des jumpers.

Programmation

N est le numéro de l'enrée à lire (0 à 7) et la variable G indique la gamme dans laquelle on veut travailler :

0 à 5 V => G = 0
-5 à +5 V => G = 4
0 à 10 V => G = 8
-10 à + 10 V => G = 12

La fonction : **XA (I2CWR1, adrmod, 128 + G + N * 16)** lance la conversion.

On utilise ensuite la fonction **I2CRD** pour lire le résultat.

Exemple :

```
XA (I2CWR1 adrmod, 140)     lance la conversion sur la première entrée en gamme -5 / +5 V
XA (I2CRD, adrmod, 2)       réalise la conversion (il n'y a pas à faire la lecture du status)
resultat = XA_getW          lit le résultat
resultat = resultat / 16    divise le résultat par 16
```

Pour les gammes -5V à 5V et -10V à +10V, si cette valeur est supérieure à 2047, faire resultat = resultat – 4095

```
If (G = 4) or (G = 12) then
    If resultat > 2047 then
        resultat = resultat – 4095
```

Enfin, la tension en millivolts est obtenue en multipliant selon la gamme :

```
If G = 0 then tension = (resultat * 5000) / 4096
If G = 4 then tension = (resultat * 10000) / 4096
If G = 8 then tension = (resultat * 10000) / 4096
If G = 12 then tension = (resultat * 20000) / 4096
```

3.4.2. Module XA-5AI

Ce module est réadressable. Il permet de lire 5 entrées analogiques entre 0 et 5 Volts en 4-20 mA. Les tensions d'entrée ne doivent pas dépasser 6 Volts par rapport à la masse (risque de détérioration du convertisseur au-delà). La précision de la conversion est de 10 bits (1/1023).

Pour utiliser une entrée en 4-20 mA, on met un jumper à la place de la barrette J1 qui correspond à l'entrée choisie. Ceci insère une résistance de précision de 250 Ohm entre l'entrée et la masse. On mesure l'intensité par la différence de potentiel aux bornes de cette résistance : pour un courant de 4 mA, on lira une tension de 1,000V soit une lecture de 205 sur une échelle de 1023. Pour un courant de 20 mA, on lira une valeur de 5 Volts, soit une lecture de 1023.

Largeur : 30 mm

Référence de tension :

Par défaut, la référence de tension est le +5Volts de l'alimentation. Ceci convient pour des mesures potentiométrique, où le capteur est également alimenté par le même +5Volts. Dans ce cas, il ne doit pas y avoir de jumper en J2.

On peut aussi utiliser une tension interne plus précise (5 Volts +/- 0.1 %). Pour cela, placer un jumper en J2. En ce cas, l'entrée 3 ne doit pas être utilisée et il ne doit pas y avoir de cavalier sur J1 en position 3. On doit alors commencer par exécuter la fonction : XA (F1, adresse).

Connexions :

Borne	1	2	3	4	5	6	7	8
	masse	Entrée 1	Entrée 2	Entrée 3 ou Vref	Entrée 4	Entrée 5	+ 5Volts	masse

Programmation :

Fonction	Adres.	arguments	Réponse	Commentaire
F1	De 9 à 127	-	S	Sélectionner référence +5V
F2		-	S	Sélctionner référence interne ou borne 3
RDCW		B	S+W	Lire l'entrée B
RDWLL		-	S+W+W+W+W+W	Lire les 5 entrées

Pour lire une entrée, utiliser la fonction XA (RDCW, adresse, num_inp) suivie de GETB (status modules) et GETW (valeur lue).
Pour lire les 5 entrées en une seule fois, utiliser XA (RDWLL, adresse) suivie de GETB (status module) et de 5 fois GETW.

Mode MINMAX

Dans ce mode, le module fait une série de mesures et retient la valeur minimale, la valeur maximale et le cumul des mesures.
La référence de tension est toujours la référence interne 5V.

On peut lancer ce mode sur 1, 2, 3 ou 4 entrées. La fréquence d'acquisition varie selon ce nombre :

Nombre	1	2	3	4
Fréquence hz)	5700	2900	2000	1500

On entre dans ce mode en spécifiant le nombre d'entrées par : XA (WRB1, adrmod, nombre)
On en sort par la fonction F1 ou F2 (voir plus haut) XA (F1, adrmod) ou XA (F2, adrmod)

L'acquisition s'arrète après 50000 acquisitions, mais on peut lire avant d'avoir atteint ce nombre.

La lecture d'une entrée se fait par la séquence suivante :

XA (RDCWLL, adrmod, entree)	où entree vaut de 0 à 3
XA_getB	lit l'état du module
XA_getw	donne le nombre de mesures effectuées
XA_getW	donne le maximum
XA_getW	donne le minimum
XA_getL	donne le cumul des valeurs sur 4 octets

La valeur moyenne s'obtient en divisant le cumul par le nombre de mesures.
La lecture relance immédiatement une nouvelle série de mesure sur l'entrée concernée.

3.4.3. Module 8 thermocouples XA-8TC

Ce module est réadressable. Il permet de lire 8 thermocouples type J ou K.
Il peut travailler sur 2 gammes :
0 à 220 degrés : placer le jumper J1 sur la carte
0 à 1200 degrés : ne pas placer le jumper J1.

Règlage de l'offset :

Après avoir choisi la gamme par le jumper J1, il faut **régler la tension d'offset**.
Pour cela, court-circuiter la première entrée et le point commun, lancer le programme Extralink pour tester le module XA-8TC, et ajuster le trimmer marqué OFS pour obtenir zéro Volts. Ne pas toucher aux deux autres potentionmètres de gain.
Le règlage d'offset est différent selon la gamme choisie.

Connexions :

Les cotés négatifs de tous les thermocouples se branchent sur la borne 1.

Borne	1	2	3	4	5	6	7	8	A	B
	Point commun	Entrée 1	Entrée 2	Entrée 3	Entrée 4	Entrée 5	Entrée 4	Entrée 6	Entrée 7	masse

Programmation :

La variable RAPPORT utilisée ci-dessous vaut 10 si on est dans la gamme 220 degrés (jumper mis) et vaut 50 dans la gamme 1200 degrés (pas de jumper).

On choisit le numéro de l'entrée (0 à 7) à lire par : XA (WRB, numero)
Attendre au moins 0,5 secondes
On lit d'abord la température de la soudure froide par : XA (RDCW, adrmod, 1)
On lit l'état du module par : XA_getB
La température froide en degrés est égale à : XA_getW * 0.4884
On lit ensuite la tension thermocouple par : XA (RDCW, adrmod, 0)
On lit l'état du module par : XA_getB
La tension de jonction est égale à : Tension = (XA_getW – 175) * RAPPORT
On fait la conversion selon le type J ou K par : XA (TJ, Tension) ou XA (TK, Tension)
La température en centièmes de degrés est égale à : XA_getL

Mode « free running »

Il est possible également de fonctionner en mode « free_running ».
Le module fait alors indéfiniment une mesure des 8 entrées thermocouples, puis fait la mesure de la soudure froide, à raison d'environ 2 mesures par seconde. Il met ainsi environ 5 secondes pour mettre à jour toutes les valeurs.

On lance ce mode par : XA (F3, adrmod)
On en sort par : XA (F4, adrmod)

On peut lire à tout moment les valeurs par : XA (F118, adrmod)
XA_getB pour lire l'état du module
8 fois XA_getW pour lire les 8 thermocouples
1 fois XA_getW pour lire la soudure froide.

3.5. Modules de sorties analogiques

3.5.1. XA-SA8

Le module XA-SA8 fournit une sortie analogique en tension 0-10 Volts ou en courant 4-20 mA sur 8 bits.
En sortie courant, le dispositif alimenté doit avoir une impédance de 500 Ohm maximum.
Deux sorties TTL et un signal PWM sont disponibles.

En sortie tension, placer le jumper J2 en position A. La sortie est sur la borne 6.
En sortie courant, placer le jumper J2 position B et connecter le dispositif 4-20 mA entre la borne 5 « 4-20low » (-) et la borne 6 « sortie AN » (+).

Ce module est basé sur un modulateur PWM à 10,8 Khz, filtré par des condensateurs. On peut choisir un filtrage plus ou moins fort en positionnant le jumper sur J1. La position A donne le filtrage le plus faible : une légère ondulation susbiste, mais les emps de montée sont rapides (constante de temps = 1 ms). La position C donne le meilleur filtrage et des temps de montée moins bons (constante de temps = 100 ms). La position intermédiaire B est conseillée.

Le signal PWM au niveau TTL est disponible sur la borne 4.

Borne->	1	2	3	4	5	6
	masse	Sortie 1	Sortie 2	PWM	4-20 low	Sortie AN

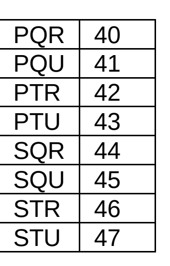
Largeur : 30 mm.

Programmation :

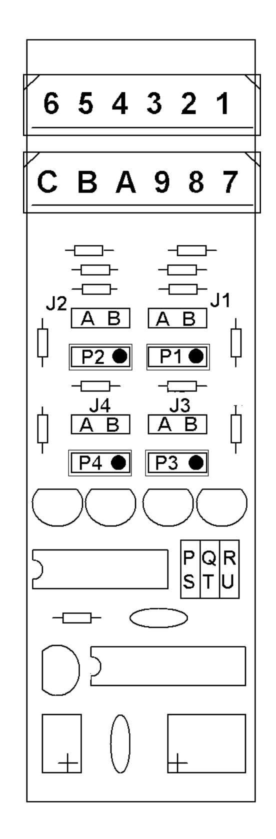
Fonction	Adres.	arguments	Réponse	Commentaire
F1	De 9 à 127	-	S	Mettre à 1 la sortie 1
F2		-	S	Mettre à 0 la sortie 1
F3			S	Mettre à 1 la sortie 2
F4			S	Mettre à 0 la sortie 2
WRB		B	S	Fixer la tension de sortie

3.5.2. Module XA-4SAB

Le module XA-4SAB fournit 4 sorties analogiques en tension 0-10 Volts ou en courant 4-20 mA sur 8 bits. Il fournit aussi une tension en 0-5 Volts, mais avec une intensité faible (< 0.1 mA).
L'adresse du module est fixée de 40 à 47 par les jumpers PQR/STU
selon le tableau suivant :



PQR	40
PQU	41
PTR	42
PTU	43
SQR	44
SQU	45
STR	46
STU	47



Pour utiliser le module en tension 0-10 Volts :
- brancher la masse en 1, 6, 7 et C
- les tensions sont disponibles sur les bornes :
3 - sortie 1
5 - sortie 2
9 – sortie 3
B – sortie 4
- les tensions 0-5 Volts sont respectivement sur 2,4,8 et A.

Pour utiliser le module en 4-20 mA, brancher la charge entre :
3 (+) et 2 (-) pour la sortie 1,
5 (+) et 4 (-) pour la sortie 2,
9 (+) et 8 (-) pour la sortie 3,
B (+) et A (-) pour la sortie4.

Le module se commande par la fonction :
XA (I2CWR2, adresse, numero, valeur)
Où **adresse** est l'adresse du module (40 à 47), **numero** est le numéro de la sortie (0 à 3) et **valeur** est la tension ou le courant généré, 0 étant le minimum et 255 le maximum.

Les potentiomètres permettent d'ajuster les tensions et courants.
Le module est livré préréglé pour une utilisation en 0-10 Volts ; il faut reprendre les règlages en mesurant la chute de tension dans une résistance pour une utilisation en courant.

3.5.3. Module XA-SA12

Le module XA-SA8 fournit une sortie analogique en tension 0->5 Volts, 0->10 Volts ou -5V -> +5V selon la position du jumper. La précision est de 12 bits.

La carte se commande par la fonction WRI, adresse, valeur.

L'argument est coompris entre 0 et 4091 ; la tension la plus basse est obtenue pour la valeur zéro.

Borne 1	Borne 2	Borne 3
Masse	Sortie analogique	2Masse

3.6. Modules de comptage

3.6.1. Module XA-UDC compteur-décompteur pour codeurs incrémentaux

Le module XA-UDC est adapté à la lecture de codeurs incrémentaux délivrant des signaux en quadrature. Le comptage se fait sur 32 bits. Les entrées des signaux de comptage doivent être aux niveaux TTL.

Le module contient également une horloge qui permet de connaître exactement le temps écoulé depuis la dernière lecture pour des mesures de vitesse.

Selon la position du jumper J1, le comptage a lieu :
- à chaque changement d'état de l'un quelconque des signaux d'entrée (jumper en AB),
- à cette fréquence divisée par 2 (jumper en BC)
- à cette fréquence divisée par 4 (pas de jumper en J1).

Les deux autres jumpers doivent toujours être installés.

Borne	1	2	3	4
	masse	entrée 1	Entrée 2	+5V

Largeur : 30 mm

Programmation :

Fonction	**Adres.**	**arguments**	**Réponse**	**Effet**
F1	De 9 à 127	-	S	Remet le compteur à zéro
RDL		-	S+L	Lit le compteur
RDLL		-	S+L+L	Lit le compteur et le temps écoulé et remet à zéro le temps

3.6.2. Module XA-CNT – Compteur rapide

Le module XA-CNT permet de compter des signaux jusqu'à 10 MHz. Il possède une horloge interne à 1/100 de seconde.

Branchements :

Borne	1	2	3
	masse	entrée CA	entrée CC

Le signal est branché sur la borne 3 s'il est aux niveaux TTL ; il est branché sur la borne 2 s'il contient une composante continue (la borne 2 est reliée à la borne 3 à travers un condensateur).

Programmation :

On définit le mode de fonctionnement parmi les suivants :
- pas remise à zéro du compteur ni de l'horloge à chaque lecture : fonction F2
- remise à zéro du compteur mais pas de l'horloge à chaque lecture : fonction F3
- remise à zéro de l'horloge mais pas du compteur à chaque lecture : fonction F4
- remise à zéro des deux à chaque lecture : fonction F5.

A tout moment, il est possible de remettre à zéro le compteur par la fonction F1.

Attention : le compteur change d'état sur chaque changement d'état du signal : transition montante et transition descendante, ce qui revient à compter à une fréquence double de celle du signal d'entrée.

On lit le compteur et l'horloge par la fonction RDLL suivie de getB (status), puis getL (compteur) et getL (horloge).

Le compteur travaille en réalité sur 64 bits. Bien que les 32 bits de poids faibles soient généralement suffisants, il est possible d'avoir accès aux 32 bits de poids forts par le fonction :
RDLLL suivie de getB (status), puis getL (compteur 32 bits forts), getL (compteur 32 bits faibles),et getL (horloge).

Programmation :

Fonction	Adres.	arguments	Réponse	Effet
F1	De 9 à 127			Remet le compteur à zéro
F2				Mode pas de remise à zero compteur ni horloge
F3				Mode remise à zero compteur et pas horloge
F4				Mode remise à zero horloge et pas compteur
F5				Mode remise zero compteur et horloge
RDLL		-	S+L+L	Lit le compteur 32 bits et le temps écoulé
RDLLL		-	S+L+L+L	Lit le compteur 64 bits et le temps écoulé

3.7. Passerelle RS-232

Le module XA-Serial permet d'échanger des messages courts avec des dispositifs série RS-232 tels que des lecteurs de codes-barres, des appareils de mesure, des terminaux d'identification.

Le module est en permanence à l'écoute du dispositif et peut travailler selon deux modes pour détecter qu'il a reçu un message :

- soit il attend un caractère de fin de message prédéfini (mode caractère de fin),
- soit il constate un silence après la reception d'une série de caractères (mode time-out).
-

Lorsque le message est reçu, il est à la disposition de l'ordinateur par l'intermédiaire du système Extralink.
Les messages ne doivent avoir plus de 30 caractères. Les vitesses vont de 1200 à 115200 Bps.

Connexions du connecteur Sub-D :

broche 2 : données émises par le dispositif vers la carte XA-serial
broche 3 : données émises par la carte XA-serial vers le dispositif
broche 5 : masse commune.

Argument	Baudrate
0	1200
1	2400
2	4800
3	9600
4	19200
5	38400
6	57600
7	115200

Programmation :

Par défaut, les valeurs sont :

- vitesse 9600 bps
- pas de parité
- mode caractère de fin
- caractère de fin = 13 (retour chariot).

Attention : le caractère 13 est souvent suivi du caractère 10 (line feed).
Il faut dans ce cas fixer 10 et non 13 comme caractère de fin.

Pour changer le caractère de fin :

XA (WRB1, adresse, codchar), où codchar est le code du caractère de fin demandé.
Ceci remet la carte en mode « caractère de fin » si elle était en mode « time-out ».

Pour mettre la carte en « mode time-out » :

XA (WRI, adresse, duree), où durée est la durée du time_out demandé.
Cette valeur est sur 16 bits ; l'unité vaut approximativement 25 µs. A 9600 bauds, chaque caractère est émis en environ 1 ms et le delai devra être au minimum de 1 caractère et demi, soit une valeur de 75. Par sécurité, mettre comme valeur 100 à 200.

On initialise le module par :

XA ('WRB', adresse, baudrate)	définit la vitesse de transmission avec le dispositif
XA (WRB2, adresse, parite)	définit le contrôle de parité (0=sans, 1=Even, 2=Odd)
XA ('F1', adresse)	active le module

Reception d'un message :

Pour lire un message envoyé par le dispositif au module, on exécute :

XA ('RDB', adresse)	demande le nombre de caractères reçus
XA_get B	donne l'état du module
Nchar = XA_getB	fournit le nombre de caractères reçus.

Un nombre plus grand que 30 indique une erreur :

245 - overrun
244 - frame
243 - message trop long
242 – un nouveau message a été reçu avant que le précédent n'ait été lu
241 – erreur de parité

Si ce nombre n'est pas nul et inférieur à 30, on lit le message par :

XA ('RDN', adrmod, nchar) lit les octets du module (nchar est le nombre de caractère obtenu plus haut)
XA_getB donne l'état du module
XA_getB donne le nombre de caractères reçus, normalement égal à Nchar.
Les octets du message sont obtenus par une suite de fonctions XA_getB.

Enfin, on relance le module pour une nouvelle réception par :
XA ('F1', adresse)

Emission d'un message :

Le message à envoyé au module est mis dans une chaine à zéro-terminal. Cette chaine ne peut donc pas contenir le caractère zéro.

Si les message sont codés en caractères ASCII, il n'y a normalement pas de problème car il n'y a ni le caractère zéro ni le caractère 255.

Si les messages sont en binaire, il se peut que la chaine contienne le caractère zéro et/ou le caractère 255.

On opère donc dans tous les cas une transcodification de la façon suivante :
- le caractère 255 sera remplacé par 2 caractères : le caractère 255 suivi d'un deuxième caractère 255.
- le caractère 0 sera remplacé par 2 caractères : le caractère 255 suivi du caractère 254.

Exemple :

10, 11, 12, 0, 13, 14, 255, 15

deviendra :

10, 11, 12, 255, 254, 13, 14, 255, 255, 15

Pour envoyer le message au module, on exécute :
XA_str (WRSTR', adresse, chaine) chaine étant un pointeur sur la chaine à zéro terminal.

4. Modules de commande de moteurs

4.1. Module XA-TMC pour moteur micropas à course réduite

Le module XA-TMC offre une solution très performante pour des moteurs jusqu'à 28 Volts - 600 mA par phase car elle réalise une commande en mode micropas (1 micropas = 1/16 de pas) et elle permet de fixer par logiciel le courant de repos et le courant de maintien. Elle comporte une entrée tout-ou-rien, mais qui n'agit pas comme fin de course.

Sa limitation principale tient en ce que les positions, étant exprimées sur 15 bits, sont limitées à 32768 micropas, soit 2048 pas, ce qui pour un moteur à 200 pas par tour, correspond à 10 tours.

Elles sont construites sur un controleur Trinamic TMC222.

Ces cartes ne sont compatibles qu'avec les bases appartenant à la version 3.1 et supérieures. De plus, la base doit recevoir la commande spéciale F3 avant de pouvoir communiquer avec les cartes XA-TMC :

If XA ('F3', 8) <> 0 then proc_erreur

Lers cartes XA-TMC ne renvoient pas l'octet de status du module et il n'y a donc pas lieu de lire la valeur du status par XA_getB après les commandes.

4.1.1. Connections

1	2	3	4
Moteur 1 phase 1A	Moteur 1 phase 1B	Moteur 1 phase 2A	Moteur 1 phase 2B

5	6	7	8
Masse alimentation	Masse switch	Switch	Alim +

4.1.2. Changement de l'adresse

Par défaut, l'adresse de la carte est 96 ou 97 selon la position du jumper. Pour mettre plus de 2 cartes cartes dans un même pack, il est possible de changer ces adresses pour 98 / 99 ou 100/101, mais ce changement est irréversible.
Pour cela, lancer le programme Extralink.exe et cliquer sur « change adress-1 » pour avoir les adresses 98/99, ou sur « change adress-2 » pour avoir les adresses 100/101. Si on clique sur « change address-2 » quand la carte a déjà été mise 98/99, les nouvelles adresses sont 102/103. Pour pouvoir utiliser un plus grand nombre d'adresses, nous consulter.

4.1.3. Programmation

Ne pas oublier d'envoyer la fonction F3 à la base (voir plus haut).
On définit les variables **octets** suivantes :

Stepmode	de 0 à 3 : mode de micropas (3 pour micropas en 1/16 de pas)
Accshape	=0 si mouvement accéléré, =1 si vitesse constante
Shaft	=0 ou 1 selon le sens de marche souhaité
Acc	acceleration de 0 à 15, généralement <= 3
Vmax	vitesse maxi de 0 à 15
Vmin	vitesse au démarrage de 0 à 15 avec Vmin < Vmax
Irun	courant en marche de 0 à 15
Ihold	courant de maintien de 0 à 15

Environ 0,5 secondes après le fin d'un trajet, le courant passe automatiquement de Irun à Ihold.

1 - Initialiser les variables, par exemple :

accshape = 0
shaft = 0
stepmode = 3
Acc = 1
Vmax = 10
Vmin = 1
Irun = 8
Ihold = 4

2 - Fixer les paramètres de fontionnement :

en exécutant les deux fonctions suivantes :

If XA ('F143', adresse, irun, ihold, stepmode, 0, shaft) <> 0 then proc_erreur
If XA ('F144', adresse, vmax, vmin, acc, accshape) <> 0 then proc_erreur

Si, par la suite, on veut ne changer qu'un paramètres (par exemple Vmax), il faut quand même exécuter ces deux fonctions.

3 - Lire le status - lire la position :

Déclarer les variables octets suivantes :

Status, i variables de travail
Def1 défauts 1
Def2 défauts 2
Motion moteur en mouvement
ESW switch fin de course

Exécuter très exactement la routine suivante :

if XA ('I2CWR1', adresse_moteur, $81) <> 0 then proc_erreur
if XA ('I2CRD', adresse_moteur, 8) <> 0 then then proc_erreur

for i = 1 to 5
Status = xa_getb
next
Def1 = Status and $7C
Def2 = Status and $0D
Status = xa_getb
motion = Status and $E0
ESW = Status and $10 ;
if (Def1 <> 0) or (Def2 <> 0) then afficher (defaut)
if (motion <> 0) then afficher (En marche) else afficher (Arreté)
if (ESW <> 0) then afficher (Fermé) else afficher (Ouvert)

IMPORTANT : La lecture du status est indispensable après la mise sous tension pour que le moteur puisse tourner. En particulier, un court-circuit ou un mauvais branchement du moteur est diagnostiqué immédiatement comme un défaut.

Le moteur ne peut pas démarrer s'il est en défaut. Il faut lire le status pour que ce défaut puisse se remettre à zéro.

Les défauts peuvent êre analysés plus finement : température, court-circuit, circuit ouvert, baisse d'alimentation, etc (se reporter au manuel du TMC222).

Pour lire la position courante, exécuter la routine suivante :

if XA ('I2CWR1', adresse_moteur, $FC) <> 0 then proc_erreur
if XA ('I2CRD', adresse_moteur, 8) <> 0 then then proc_erreur
position = xa_getb
posh = xa_getw

4 - Lancer le moteur

<Objectif> est une variable comprise entre -16383 et 16384 :

If XA ('F141', adresse, $8B, objectif)) <> 0 then proc_erreur

Arret immédiat sans décélération : **XA ('I2CWR1', adresse, $85)**
Arret immédiat après décélération : **XA ('I2CWR1', adresse, $8F)**

Attention : cette fonction met le contrôleur en défaut.

5 - Définition de la position origine

On remet à zéro la position courante du moteur pas la fonction :

XA ('I2CWR1', adresse, $86)

4.2. Cartes pour moteurs pas à pas micropas XA-MSM et XA-M3SM

Caractéristiques générales

Les modules XA-MSM et XA-M3SM permettent de commander des moteurs pas à pas en micropas jusqu'à 1,5 Ampère par phase. Elles réalisent les mouvements avec accélérations et décélérations et elles peuvent réguler et limiter l'intensité du courant dans les moteurs. Elle reçoivent une entrée tout-ou-rien par moteur, qui agit comme interrupteur de fin de course.

Le module XA-MSM permet de commander un seul moteur et le module XA-M3SM permet de commander 2 ou 3 moteurs simultanément et indépendamment. Il est équipé d'un driver pour les deux premiers moteurs et on peut y ajouter un troisième driver de puissance si l'on souhaite commander trois moteurs. Les drivers sont montés sur porte-circuits et sont facilement enfichables sur la carte.

La tension d'alimentation doit être comprise entre 7 et 28.5 Volts.

Logiciel.

Le logiciel utilise entre autres les fonctions SPI32 et XA-GET24, qui sont disponibles seulement dans les librairies XADLL d'indice supérieur ou égal à 3.2.

La commande SPI32 permet d'écrire un mot de 32 bits dans l'un des registres du contrôleur du moteur (seuls les 24 bits de poids faibles sont pris en compte), et elle retourne des informations sur le contrôleur.

Paramétrage du contrôleur.

Chaque moteur est normalement sous le sontrole d'un seul fin de course qui correspond à l'extrémité « gauche » de sa course. Si on utilise qu'un seul moteur, on a la possibilité d'utiliser le fin de course du 3eme moteur pour agir à l'autre extrémité. Dans ce cas et dans ce cas seulement, on envoie en tout premier lieu la fonction :

XA ('WRCB', adrmod, 11, 2)

On indique au contrôleur le nombre de drivers qui sont installés par la fonction :

XA ('WRCB', adrmod, 0, N) ou N est le nombre de moteurs.

Par défaut le nombre de moteurs est égal à 1.

Chaque moteur est paramétré indépendamment des autres. Pour indiquer à quel moteur on s'adresse, on utilise la fonction :

XA ('WRB', adrmod, N) où N est compris entre 1 et 3 pour la carte XA-M3SM.

Il est inutile d'utiliser ces fonctions pour la carte XA-MSM, dont le seul moteur est assigné par défaut.

Paramétrage d'un moteur

Le controleur peut agir selon 4 modes différents, qui sont détaillés plus loin. Le mode par défaut est le mode RAMPE, qui réalise un profil de vitesse trapézoidal, avec des accélérations et décélérations constantes.

La base de temps agit directement sur l'horloge liée à un moteur. On la fixe en exécutant deux fonctions :

XA ('WRCB', adrmod, 8, DIV), puis **XA ('WRCB', adrmod, 9, DIV)**, où DIV est compris entre zéro et cinq.

Le nombre de microspas est fixé par la fonction :

XA ('WRCB', adrmod, 10, NMP) où 0 <= NMP < 6, la valeur 6 correspondant à 64 micros pas par pas entier.

On fixe ensuite les paramètres habituels de fonctionnement :

Accélérations : **XA ('SPI32', adrmod, 6, ACC) où 0 < ACC < 2048**
Vitesse maxi : **XA ('SPI32', adrmod, 3, VMAX) où 0 < VMAX < 2048**
Vitesse de départ : **XA ('SPI32', adrmod, 2, VMIN) où 0 < VMIN < 2048**

On fixe enfin les **courants dans les moteurs**. Il est possible de fixer indépendamment le courant pour des fortes accélérations, le courant pour des accélérations plus faibles et le courant de repos.

L'accélération, nommée ACCTH, à partir de laquelle on souhaite un courant plus important est fixée par la fonction :

XA ('SPI32', adrmod, 8, ACCTH) où 0 < ACCTH < 2048.

Le courant (fort) pour des accélérations supérieures à cette valeur est fixé par :
XA (WRCB', adrmod, 1, IACC) où 0 < IACC < 8.

Le courant (normal) pour des accélérations plus faibles et la marche à vitesse constante est fixé par :
XA (WRCB', adrmod, 2, IRUN) où 0 < IRUN < 8.

Enfin le courant (faible) de repos ou courant de maintien (mouvement terminé) est fixé par :
XA (WRCB', adrmod, 3, IHOLD) où 0 < IHOLD < 8.

On peut à tout moment fixer la position courante par la fonction :
XA ('SPI32', adrmod, 1, CPOS) où 0 < CPOS < 8388607 (=$7FFFFF).

Les positions sont exprimées en micro pas sur 24 bits. Elles peuvent être positives ou négatives. À l'initialisation, la position courante est égale à zéro.

Une fois ces différentes valeurs fixées, on lance le moteur vers une position par la fonction :
XA (SPI32', adrmod, 0, OBJ)
où OBJ est la position objectif, positive ou négative, sur 24 bits.

Note : tous les paramètres sont établis par défaut à des valeurs correctes à l'initialisation. Il suffit donc généralement, après avoir déclaré le nombre de moteurs, de fixer un objectif positif et non nul (par exemple 50000) pour que le moteur tourne avec les paramètres par défaut.

Modes de fonctionnement

Les 4 modes de fonctionnement sont fixés par la fonction :
XA ('WRCB', adrmod, 7, mode) où 0 <= mode <= 3
0 : mode RAMPE, avec acceleration et décélération égales et constantes
1 : mode SOFT, avec une décélération plus douce
2 : mode SPEED ou VELOCITY, à vitesse constante
3 : mode HOLD, dans lequel le moteur maintient sa vitesse.

Les modes normalement utilisés sont les modes 0 et 2.

En mode Velocity (2), le moteur démarre et tourne indéfiniment à la vitesse fixée par :
XA ('SPI32', adrmod, 4, VTARGET) où 0 < VTARGET < 2048

Le fait de passer du mode 0 ou 1 au mode 3 en cours d'accélération ou de décélération garde la vitesse qu'avait le moteur à ce moment. Le fait de repasser au mode 0 provoque la reprise de l'accélération ou l'arrêt du moteur selon le point qu'il a atteint. S'il a dépassé l'objectif, il s'arrête mais ne revient pas en arrière.

Lectures

La commande SPI32 retourne le statut du contrôleur sur 1 octet, qu'on obtient par XA_getB, et la valeur qu'il y avait dans le registre, qu'on obtient par XA_get24.

Les variables New_value et Old_value étant des entiers sur 32 bits, on écrira :

```
if XA ('SPI32', adrmod, registre, new_value) <> 0 then traitement_erreur_base
if XA_getB <> 0 then traitement_erreur_module
status_controleur = XA_getb
old_value = XA_get24
```

Si on veut connaître la valeur qui est contenue dans un registre sans le modifier, on utilisera la fonction RDCL. Attention : cette fonction doit être exécutée deux fois de suite, la première fois ayant pour effet de demander la lecture et la seconde de transférer le résultat de la lecture.

Par exemple, pour lire la position, on écrira :

XA ('RDCL', adrmod, 1)	une première fois
XA ('RDCL', adrmod, 1)	une seconde fois
XA_getb	pour connaître l'état du module, et
XA_get24	pour avoir la valeur fixée pour l'accélération.

L'octet status_controleur permet pour chaque moteur de savoir s'il a terminé et quel est l'état du switch correspondant. Il se décompose de la manière suivante :

bit 0 : 1 si le moteur 1 tourne, 0 sinon
bit 1 : 1 si le switch 1 est fermé, 0 sinon
bit 2 : 1 si le moteur 2 tourne, 0 sinon
bit 3 : 1 si le switch 2 est fermé, 0 sinon
bit 4 : 1 si le moteur 3 tourne, 0 sinon
bit 5 : 1 si le switch 3 est fermé, 0 sinon

L'État du driver d'un moteur se lit par la fonction RDCI. La variable N étant le numéro du moteur de 1 à 3, et Status_controleur étant un mot de 16 bits, on écrira :

if XA ('RDCI', adrmod, N) <> 0 then traitement_erreur_base
if XA_getB <> 0 then traitement_erreur_module
status_driver = XA_getW

Cette variable se décompose comme suit :

bit 0 : Overcurrent low A side
bit 1 : Overcurrent low B side
bit 2 : No load A side
bit 3 : No load B side
bit 4 : Overcurrent PWM
bit 5 : Low supply voltage
bit 6 : Temperature prewarning
bit 7 : Over temperature

Fins de course

Par défaut, à l'initialisation, le fin de course « gauche » agit sur le moteur et, s'il est ouvert, il empêche le moteur d'aller dans le sens négatif (vers une position inférieure à la position actuelle). En cours de mouvement, le fin de course arrête brusquement le moteur s'il s'ouvre. On peut désactiver l'action du fin de course gauche sur le moteur en écrivant

XA ('WRCB', adrmod, 6, xfdc) où xfdc est un octet dont :

Le bit 0 vaut 1 pour inhiber le 1er fin de course
Le bit 1 vaut 1 pour inhiber le 2eme fin de course (fin de course droit, dans le cas d'un seul moteur, si initialisation adequate)
Le bit 2 vaut 1 pour obtenir une décélération plutôt qu'un arrêt brusque sur les fins de course.

Connexions

Les drivers des moteurs sont montés sur des connecteurs enfichables. Respecter impérativement l'orientation des circuits, qui est indiqué par le rond blanc noté "PIN1".

Sur la carte XA-M3SM, les contrôleurs sont obligatoirement installés dans l'ordre 123 : il ne peut pas y avoir de contrôleur en 2 si il n'y a pas un contrôleur en 1. Lorsqu'un seul contrôleur est installé, un cavalier doit être positionné en J5 et aucun cavalier en J10. Si deux contrôleurs sont installés, un cavalier doit être installé en J10 et aucun en J5. Enfin, si les trois contrôleurs sont installés, aucun cavalier ne doit être positionné ni en J5 ni en J10.

L'alimentation du moteur doit être reliée sur les bornes 1 (moins) et 2 (plus). Le moins de cette alimentation est commun avec la masse du système.

Connexion des moteurs pour la carte M3SM

Sur la carte M3SM, chaque moteur est assigné à un connecteur différent selon le nombre de drivers qui sont installés.

1 driver installé : cavalier en JP5 et connections en 3456

3	4	5	6
Moteur 1 A	Moteur 1 A	Moteur 1 B	Moteur 1 B

2 driver installés : cavalier en JP10, moteur 1 en 789A et moteur 2 en 3456

3	4	5	6
Moteur 2 A	Moteur 2 A	Moteur 2 B	Moteur 2 B
7	8	9	A
Moteur 1 A	Moteur 1 A	Moteur 1 B	Moteur 1 B

3 driver installés : pas de cavalier, moteur 1 en BCDE, moteur 2 en 789A et moteur 3 en 3456.

3	4	5	6
Moteur 3 A	Moteur 3 A	Moteur 3 B	Moteur 3 B
7	8	9	A
Moteur 2 A	Moteur 2 A	Moteur 2 B	Moteur 2 B
B	C	D	E
Moteur 1 A	Moteur 1 A	Moteur 1 B	Moteur 1 B

Connexion des fins de course

Masse fdc	Fdc moteur 1	Fdc moteur 2	Fdc moteur 3
F	G	H	I

Les switches doivent normalement établir un contact avec la masse (borne 3). Ce contact doit être coupé lorsque le moteur atteint leur position.

Connexion des moteurs pour la carte MSM

1	2	3	4	5	6
Moins alim	Plus alim	Moteur 1 A	Moteur 1 A	Moteur 1 B	Moteur 1 B

Connexion des fins de course

Masse fdc	Fdc moteur 1	Fdc moteur 2	Fdc moteur 3
7	8	9	A

Les switches doivent normalement établir un contact avec la masse (borne 3). Ce contact doit être coupé lorsque le moteur atteint leur position.

Règlage de l'itensité.

L'intensité maximum dans les bobinages se règle avec les séries de 4 cavaliers situés sous les drivers.

J1	J2	J3	J4	Moteur 1
J6	J7	J8	J9	Moteur 2
J11	J12	J13	J14	Moteur 3
Non	Non	Non	Non	250 mA
Oui	Non	Oui	Non	500 mA
Non	Oui	Non	Oui	1030
Oui	Oui	Oui	Oui	1550 mA

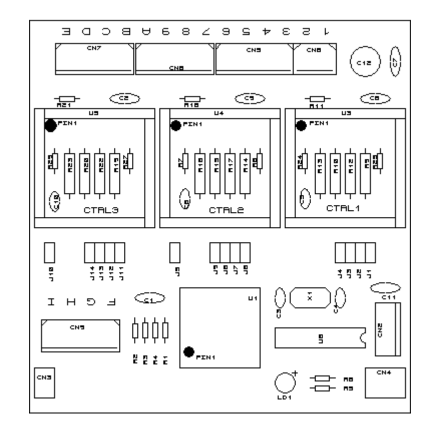
Récapitulatif des fonctions utilisées

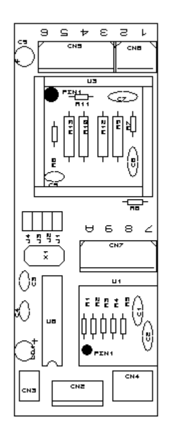
SPI32	
0	Objectif
1	Position actuelle
2	V. mini
3	V. maxi
4	V. objectif (mode velocité)
5	(Read only) V. réelle
6	Acceleration
7	(Read only) Acc. Réelle
8	V. seuil
13	Tolerance index
14	Position index

WRCB	
1	Courant fort
2	Courant moyen
3	Courant faible
4	Pmul (non utilisé)
5	Pdiv (non utilisé)
6	Inhibition switches fdc
7	Mode vitesse
8	Pulsediv (base de temps)
9	Rampdiv (base de temps)
10	Micropas
11	Mode fin de course

Valeurs par défaut à la mise sous tension

Nombre de moteurs : 1
Objectif = 0
Position = 0
v.mini = 1
v.maxi = 1000
acceleration = 100
courant fort = 3
courant moyen = 3
courant faible = 3
inhibition = 0 (fins de course actifs)
base de temps = 5
micropas = 5 (1/32)





4.3. Modules XA-GSD et XA-298 pour moteurs pas à pas

4.3.1. Présentation

Les modules de commande de moteurs pas à pas peuvent fonctionner selon 3 modes :
- mode trajet à vitesse constante
- mode progressif avec accélération / décélération
- mode vitesse constante indéfinie

Ils travaillent en mode demi-pas et donnent donc 400 positions par tour pour des moteurs de 200 pas par tour.

Les sorties de puissance sont adaptées aux moteurs bipolaires.

Dans tous les modes, la base de temps définit la vitesse de l'horloge. Elle peut avoir 4 valeurs :
1 = vitesse d'horloge / 1 (la plus rapide)
2 = vitesse d'horloge / 2
3 = vitesse d'horloge / 4
4 = vitesse d'horloge / 8 (la moins rapide).

Deux entrées de fin de course permettent de stopper immédiatement le moteurs si elles sont fermées (mises à la masse). Chaque entrée correspond à un sens de rotation et ne stoppe le moteur que s' il va dans ce sens. On doit donc s'assurer que le moteur est bien arrêté par le fin de course vers lequel il se dirige. Les arrêts sur fins de course peuvent être inhibés par les fonctions F7 et F8.

L'état du module peut être lu par la fonction RDI suivi de 2 fois GETB. Il est donné dans le premier octet :
0 : arrêt ; 1 : en accélération ; 2 = en palier ou vitesse constante ; 3 = en décélération ; 4 = en marche indéfinie ; 5 = arret sur fin de course. L'état des fins de course est donné dans le second octet par les deux bits de poids faibles.

F	A	R	Fonctions d'écriture
WRB	B	S	Fixer DSI (1..4)
WRI	W	S	Fixer période en mode vitesse constante (16 bits)
WRB1	B	S	Fixer rampe d'accélération (0..254)
WRB2	B	S	Fixer taux d'accélération (0..7)
WRL	L	S	Fixer course / palier (32bits)
F1	-	S	Démarrage mode vitesse constante sens 1
F2	-	S	Démarrage mode vitesse constante sens 2
F3	-	S	Démarrage mode progressif sens 1
F4	-	S	Démarrage mode progressif sens 2
F5	-	S	Démarrage sans limite sens 1
F6	-	S	Démarrage sans limite sens 2
F7	-	S	Activer FDC
F8	-	S	Désactiver FDC
F9	-	S	Arret en décélération
F10	-	S	Arrêt brusque
F11	-	S	Rétablir alimentation
F12	-	S	Couper alimentation
RDI	-	SBB	Lecture etat moteur et fins de course (note 1)
RDL	-	SL	Lecture course (32 bits)

4.3.2. Commande du mode "Vitesse Constante"

Pour effectuer un trajet à vitesse constante, on doit :
- fixer la période entre deux pas par la fonction WRI. Plus la période est élevée, plus la vitesse est lente.
- Fixer la course par la fonction WRL
- Lancer le moteur par les fonctions F1 ou F2.

4.3.3. Mode "progressif" - principes de base

Le mode progressif fonctionne selon les principes suivants :

1 - Le nombre de pas de la phase accélération est égal à celui de la phase décélération. On appellera cette variable NPAD.

2 - Les phases accélération et décélération sont composées d'un nombre d'étapes égal à RAMPE.

3 - Chaque étape est composée d'une nombre de pas égal à TACC qu'on appelera taux d'accélération. L'accélération la plus forte sera donc obtenue avec TACC=1, et on aura des accélérations plus douces si on augmente la valeur deTACC. Elle doit être comprise entre 1 et 7. Sa valeur par défaut est égale 1.

L'accélération comprend donc un nombre de pas égal à TACC multiplié par <RAMPE>.

4 - La vitesse obtenue est fonction du nombre d'étapes d'accélération : plus on augmente la cette valeur (RAMPE), plus la vitesse est élevée. La valeur maximale de RAMPE est égale à 254.

6 - Entre la phase d'accélération et la phase de décélération, il peut y avoir une phase à vitesse constante. Le nombre de pas à vitesse constante sera appelé <PALIER>, il peut éventuellement être nul.

Le tableau suivant donne les rapports de vitesse obtenus selon le nombre d'étapes (RAMPE) :

Rampe	Vitesse	Rampe	Vitesse	Rampe	Vitesse
254	100	149	60	40	20
230	90	120	50	26	15
204	80	95	40	12	10
176	70	67	30	2	6

4.3.4. Mode progressif - algorithme

Pour programmer un déplacement d'un nombre de pas total égal à COURSE, compte tenu d'un taux d'accélération fixé par TACC, on suivra les étapes suivantes :

1 - dans la table, trouver la valeur de RAMPE pour la vitesse souhaitée. **Attention** : la rampe ne doit pas etre supérieure à 254.

2 - calculer le nombre de pas d'accélération-décélération = NPAD = 2 x TACC x RAMPE

3 – on doit réduire la longueur de la rampe si la course est trop courte :
  si NPAD < COURSE alors
   RAMPE_RELLE = RAMPE
 sinon
   RAMPE_REELLE = COURSE / (2 * TACC)

4 - la rampe doit être au moins égale à 2 :
Si RAMPE_REELLE < 2
alors RAMPE_REELLE = 2

5 - recalculer
  NPAD = 2 * TACC * RAMPE_REELLE

6 - Si NPAD est inférieur ou égal à la course totale, alors :
- le nombre de pas du palier est :
PALIER = COURSE - NPAD
- le nombre de pas de la rampe est : RAMPE_REELLE

Si NPAD est strictement supérieur à la course totale, il faut utiliser la fonction de mouvement à vitesse constante, car le mouvement est trop court pour pouvoir faire accélération et décélération.

Fixer le nombre de pas de la rampe par la commande WRB1 ; sa valeur est comprise entre 1 et 255.
Fixer le taux d'accélération par la commande WRB2 ; sa valeur est comprise entre 1 et 4.
Fixer le palier par la commande WRL.
Lancer le moteur par la commande F3 ou la commande F4.

4.3.5. Mise en service

Dans un premier temps, on pourra utiliser les valeurs suivantes :
- NACC = 1
- base de temps = 4
- course = 5000
- période = 1000

4.3.6. Problèmes possibles

- le moteur vibre, mais ne tourne pas: mauvais branchement du moteur à la carte de puissance, probablement une inversion de deux fils.

- le moteur ne démarre pas en mode progressif : voir si le moteur n'est pas arrêté sur un fin de course. La lecture de l'état des fins de course doit donner la valeur 3. Sinon, s'écarter du fin de course en mode vitesse constante.

4.3.7. Module XA-298 et XA-GSD – branchements

Largeur : 40 mm

Brancher l'alimentation du moteur entre les bornes 1 (Masse) et 8 (Alim +).
Brancher un fin de course entre la borne 1 (masse) et la borne 2, l'autre entre la borne 1(masse) et la borne 3.
Une phase du moteur se branche entre les bornes 4 et 5, l'autre entre les bornes 6 et 7.

Borne	1	2	3	4	5	6	7	8
	masse	Fdc 1	Fdc 2	Moteur 1A	Moteur 1B	Moteur 2A	Moteur 2B	Alim +

4.4. Modules d'asservissement de moteurs courant continu

4.4.1. Présentation des modules XA629 et XA-629CS

Les modules XA-629 et XA-629CS commandent un moteur courant continu en asservissement de vitesse ou en asservissement de position. Chaque moteur connecté à un module XA constitue ainsi un actionneur autonome doué de sa propre intelligence, capable d'interpréter des ordres de haut niveau (accélérations, déplacements, demandes de position, etc...)

L'asservissement est réalisé par un contrôleur spécialisé LM629 (National Semiconductor), qui est documenté par les notes d'application AN 706 et AN693 de ce fabricant. Les module reçoivent l'information sur les mouvements du moteur par un codeur incrémental qui fournit des signaux en quadrature.

Le controleur multiplie par 4 la résolution du codeur en analysant chaque transition des sinaux d'entrée : un codeur à 500 points par tour donnera en fait une résolution de 2000 points.

Le module XA-629 intègre la commande de puissance et est adapté à des moteurs fonctionnant sous 12 à 48 Volts CC avec une intensité maximale de 3 A. La carte XA-629CS doit être utilisée avec un pont en H tel que la carte de puissance XA-GPC pour commander des moteurs fonctionnant sous 24 à 48 V avec une intensité maximale de 10 A.

Largeur : XA-629 : 50 mm
XA-629CS : 40 mm.

4.4.2. Asservissement de position et asservissement de vitesse

Dans le mode **asservissement de position**, on spécifie l'accélération, la vitesse maximale et la position à atteindre. Le moteur effectue le mouvement en respectant les consignes : il accélère jusqu'à atteindre le vitesse maximale ou jusqu'à ce qu'il doive commencer la décélération pour atteindre la position finale. Le taux de décélération est égal au taux d'accélération.

Dans le mode **asservissement de vitesse**, le moteur accélère jusqu'à la vitesse spécifiée avec l'accélération spécifiée, et maintient cette vitesse jusqu'à ce qu'on la modifie ou qu'on envoie un ordre d'arrêt.

On ne doit pas fixer une vitesse ou une accélération supérieures à celles que le moteur peut physiquement atteindre. Il est bon de garder une marge, et de me pas dépasser environ 80% des performances maximales du moteur.Il est possible à tout moment de lire le status du moteur, qui indique notamment si le moteur a terminé son trajet et s'il y a eu un effort mécanique anormal.

4.4.3. Régulation PID

Le contrôleur utilise un filtre logiciel digital PID (proportionnel intégral différentiel) pour calculer la consigne envoyé au moteur.

Celle-ci est calculée en permanence en faisant la somme de 3 facteurs :
1 - l'écart de position
2 - l'intégrale dans le temps de cet écart
3 - la dérivée de la position (la vitesse).

Le premier facteur génère une force de compensation proportionnelle à l'écart de position, exactement comme le ferait un ressort. Son coefficient est **KP.**

Le second facteur (intégral) fournit une force qui augmente avec le temps et l'écart de position. Il permet d'affiner la position finale et de contrecarrer une éventuelle force permanente exercée sur le moteur à l'arrêt. Il assure ainsi que l'erreur de position statique est nulle.

Ce terme peut être limité de façon à réduire son influence dans le processus d'asservissement. Son coefficient est **KI**, la limite qu'on lui assigne est **IL**

Le troisième facteur, (dérivée) fournit une force d'amortissement proportionnelle aux taux de variation de l'écart de position. Elle agit comme une force de frottement visqueux. Son coefficient est **KD**.

L'intervalle de temps associé au calcul de la dérivée peut être fixé par l'utilisateur : en général, des intervalles de temps plus longs sont nécessaires pour les opérations à basse vitesse. L'intervalle d'échantillonnage est déterminé par le coefficient **DSI**.

Les valeurs de KP, KI, KD, IL et DSI doivent être bien adaptées expérimentalement à la réponse mécanique du système. Il est alors possible de contrôler une très large gamme de moteurs et de charges d'inertie avec un mouvement régulier exempt de vibrations, même à très basse vitesse.

4.4.4. Limitation du couple

Il est possible de fixer une limite à l'erreur de position admissible lors d'un trajet. Ceci permet de détecter un blocage, des efforts anormaux sur le moteur, ou plus généralement, l'impossibilité d'atteindre l'objectif fixé compte tenu de la vitesse et de l'accélération demandées. Aux basses vitesses, l'écart de position est sensiblement proportionnel au couple exercé par le moteur.

Fixer la limite de l'écart de position revient donc à fixer le couple maximum.
Avant de lancer un mouvement, on doit spécifier au contrôleur ce qu'il devra faire s'il rencontre cette limite. Il y a deux cas possibles :
- il coupe immédiatement l'alimentation du moteur (commande LPES),
- il signale seulement cette anomalie, tout en continuant de s'efforcer d'asservir (LPEI).

On évitera d'utiliser cette dernière possibilité, car elle peut engendrer un échauffement excessif de la carte et du moteur.

4.4.5. Calage sur l'index

A la mise sous tension, il est généralement nécessaire de connaître la position zéro, qui est dite position d'index. Il est souvent suffisant de faire venir le moteur en butée, avec un vitesse réduite et une limite de couple faible, et de lire la position atteinte au moment du blocage.

Pour plus de précision, la carte permet de gérer un interrupteur ou détecteur de position, généralement utilisé en fin de course, dont l'ordinateur peut lire l'état à tout moment. Si le contrôleur a reçu la commande de déverrouillage de l'index (SIP), il enregistrera dans son registre index la position au moment exact où a lieu le contact du détecteur. Ceci permet d'effectuer un calage mécanique très précis. Le détecteur de fin de course doit mettre à la masse l'entrée FDC.

4.4.6. Arrêt en cours de trajet

Il y a trois possibilités pour stopper le moteur au cours d'un déplacement :

1 - Débrayage : le moteur n'est plus alimenté et il termine sa course "en roue libre".
2 - Décélération : le moteur entame immédiatement la décélération de fin de parcours.
3 - Blocage : le moteur est alimenté en sens inverse pour réaliser le freinage le plus bref possible.

Ce dernier type de freinage crée des efforts mécaniques et génère de forts appels de courant. Il est à éviter, si ce n'est avec des moteurs de faible puissance ou si le moteur tourne à petite vitesse.

Lorsque le moteur est "débrayé", il peut être tourné à la main. Le contrôleur continue à tenir à jour la position réelle à partir des signaux que lui envoie le codeur, ce qui permet le repérage de positions pour apprentissage.

4.4.7. Module XA-629 - branchements

Borne	1	2	3	4	5	6	7	8	9	A
	Masse codeur	Signal 1 codeur	Signal 2 codeur	+5V codeur	Moteur 1	Moteur 2	Index	Entrée TOR	Alim + moteur	Alim - moteur

Le codeur se branche sur les bornes 1, 2, 3 et 4 (s'il est alimenté en 5 Volts).
Le détecteur de fin de course se branche sur Index. Une entrée tout-ou rien est disponible sur la borne 8. Ces deux entrées sont munies de pull-up et le signal doit donc mettre l'entrée à la masse lorsqu'il y a contact.
Le moteur se branche sur les bornes 5 et 6.
L'alimentation destinée au moteur se connecte entre Alim- moteur et Alim+ moteur (bornes 9 et A).

4.4.8. Module XA-629CS + carte XA-GPC - branchements

Module XA-629CS :

Borne	1	2	3	4	5	6	7	8	9	A
	Masse codeur	Signal 1 codeur	Signal 2 codeur	+5V codeur	Index	Entrée TOR	Inh	Dir	PWM	masse

Le codeur se branche sur les bornes 1, 2, 3 et 4 (s'il est alimenté en +5 Volts).
Le détecteur de fin de course se branche sur Index. Une entrée tout-ou rien est disponible sur la borne 8. Ces deux entrées sont munies de pull-up et le signal doit donc mettre l'entrée à la masse lorsqu'il y a contact.
Les bornes notées MASSE, DIR (sens de rotation), PULSE (impulsions de commande) et INH (inhibition) sont à relier aux bornes correspondantes de la carte de puissance. Le signal INH inhibe le pont s'il est haut.

Carte XA-GPC :

La carte XA-GPC est électriquement isolée par optocouplage de la carte XA-629CS et donc du bus Extralink. Elle nécessite une alimentation exterieure dont la tension est entre 12 et 48 VCC.

Le tableau ci dessous montre les connexions avec le moteur, l'alimentation et la carte XA-629CS (3 eme ligne du tableau).

Borne	1	2	3	4	5	6	7	8
XA-GPC	masse	PWM	dir	inh	moteur	moteur	Alim +	Alim -
XA-629CS	Borne A	Borne 9	Borne 8	Borne 7	-	-	-	-

4.4.9. Branchements du codeur et du moteur

Nous recommandons l'utilisation des codeurs Hewlett-Packard série HEDS-55xx montés directement en bout d'arbre moteur. Le codeur ne doit avoir aucun jeu mécanique, qui engendrerait des instabilités dans l'asservissement.

Il doit y avoir **concordance** entre le sens du moteur et le sens du codeur : une consigne de rotation positive doit aller vers une valeur de position plus grande. Si ce n'est pas le cas, le contrôleur augmente d'autant plus son erreur de position qu'il tente de se rapprocher de l'objectif. Il accélère très brutalement, et atteint presque immédiatement l'erreur limite.

Ce phénomène indique une opposition de sens entre le moteur et le codeur. Il suffit alors d'intervertir les deux fils du moteur, ou bien les deux phases du codeur.

4.4.10. Coupure de l'alimentation moteur

Les cartes XA-629 et XA-629CS permettent de couper l'alimentation du moteur ("débrayer") par la commande PDBR afin de pouvoir le bouger manuellement. Cependant, le moteur reste électriquement court-circuité et il peut être relativement dur à tourner, particulièrement s'il est équipé d'un réducteur. C'est pourquoi il est possible de libérer électriquement le moteur. Pour cela, utiliser la fonction 12 pour couper totalement l'alimentation et la fonction 11 pour la rétablir (voir la table ci-dessous).

4.4.11. Mode Autostop

A partir de la version 1.3, les cartes XA-629 et XA-629CS ont la possibilité de couper automatiquement l'alimentation du moteur si l'une quelconque des entrées (fin de course) est mise à la masse. Cette action est équivalente à la fonciton F10. On doit ensuite exécuter la fonction F11 pour pouvoir relancer le moteur.

Pour activer le mode Autostop, exécuter la fonciton F13. Pour le désactiver, fonction F14. Par défaut, le mode Autostop n'est pas actif.

4.4.12. Liste des commandes

- **lecture du status**

A chaque fonction, le status du module qui est renvoyé dans la réponse apporte les informations suivantes :
- si le bit 7 est à 1, il y a eu arrêt sur limite
- si le bit 6 est à 1, le trajet est fini,
- si le bit 5 est à 1, une commande incorrecte a été envoyée au controleur
- si le bit 4 est à 1, l'index a été trouvé.

Le status est remis à son état initial par la commande **RSTI** (reset anomalies). Par exemple, lorsqu'il y a eu limite atteinte, il faut envoyer la fonction **RSTI** avant de relancer ensuite le moteur vers un autre objectif par : XA ('WRL3', adrmod, objectif).

Il est recommandé de s'assurer après chaque commande qu'elle a été bien comprise (bit 5 du status module à zéro).

Fonction	Argument	Réponse	Fonctions d'écriture
F1	-	S	RST – reset controleur
F2	-	S	RSTI – reset anomalies
F3	-	S	SIP - lancer acquisition index
F4	-	S	DFH – définir position d'origine
F5	-	S	PBKR – arret par décélération
F6	-	S	PBLK – arrêt par blocage
F7	-	S	PDEBR – arrêt en « roue libre » (débrayage)
F8	-	S	Démarrer asservissement de vitesse sens 1
F9	-	S	Démarrer asservissement de vitesse sens 2
F10	-	S	Couper alimentation
F11	-	S	Rétablir alimentation
F12	-	S	Démarrage
F13	-	S	Activer mode Autostop
F14	-	S	Désactiver mode Autostop
WRI	-	S	LPEI – Fixer limite couple, l'asservissement continue si atteinte
WRI1	-	S	LPES – fixer limite de couple, cessation de l'asservissement si atteinte
WRL	L	S	Fixer l'accélération
WRL1	L	S	Fixer la vitesse
WRL2	L	S	Fixer l'objectif
WRL3	L	S	Fixer l'objectif + lancement
SET629	B+W+W+W+W	S	Fixer DSI, KP, KD, KI, IL + validation
RDB	-	S+B	Lire entrées
RDI	-	S+B+B	Lire entrées (le 1er octet seulement est significatif)
RDCI	1	S+W	Lire vitesse
RDCI	2	S+W	Lire intégrale
RDCL	1	S+L	Lire position
RDCL	2	S+L	Lire index
RDCL	3	S+L	Lire vitesse théorique
RDCL	4	S+L	Lire position théorique

Exemple de programmation

Pour faire touner un moteur, envoyer la séquence de commandes suivantes :

XA ('F11', adrmod)	Rétablir alimentation si elle avait été coupée
XA ('F1', adrmod)	Reset controleur
XA ('F2', adrmod)	Reset anomalies
XA ('WRL', adrmod, acceleration)	Fixer acceleration = 200
XA ('WRL1', adrmod, speed)	Fixer vitesse = 200 000
XA ('SET629', adrmod, 2, 100, 100, 100, 100)	Fixer kp kd ki il dsi
XA ('WRI1', adrmod, limite)	Fixer limite = 200
XA ('WRL3', adrmod, objectif)	fixer objectif et lancer

4.5. Module XA-CCM moteur CC en boucle ouverte

Le module XA-CCM commande un moteur courant continu en boucle ouverte en fonction d'interrupteurs de fin de course et d'interrupteurs d'index intermédiaires, avec la possibilité de fixer la vitesse.

Les interrupteurs de fin de course ou d'index intermédiaire doivent faire contact lorsque la position correspondante est atteinte.

Puissance moteur : 12 à 48V – 3A.

4.5.1. Branchements

Brancher l'alimentation du moteur entre les bornes 7 (Alim +) et 8 (Alim -).
Brancher un fin de course entre la borne 1 (masse) et la borne 2, l'autre entre la borne 1(masse) et la borne 4.
Les interrupteurs d'index intermédiaires doivent être branchés en parallèle entre la masse (borne 1) et la borne 3.

Borne	1	2	3	4	5	6	7	8
	masse	Fdc 1	Index(s)	Fdc2	moteur	moteur	Alim +	Alim -

4.5.2. Programmation

Fonction	Adresse	arguments	Réponse	Effet
F1	De 9 à 127	-	S	Démarrage sens 1 sans limite
F2		-	S	Démarrage sens 2 sans limite
F3		-	S	Démarrage sens 1 limite sur fin de course
F4		-	S	Démarrage sens 2 limite sur fin de course
F5		-	S	Démarrage sens 1 limite sur index
F6		-	S	Démarrage sens 2 limite sur index
F7		-	S	Arret freinage par courant de Foucault (bobinage court-circuité)
F8		-	S	Arret par coupure de l'alimentation
WRB		B	S	Fixer la vitesse (0 à 255)
RDB		-	S+B	Lire l'état des entrées fin de course et index, status moteur

La fonction RDB renvoie un octet :
les trois bits de poids faibles indiquent l'état des fins de course et des index
le bit de pids fort est à 1 si le moteur tourne.

Remarque : avant de lancer le moteur, il faut lui fixer sa vitesse par la fonction XA (20, adresse, vitesse).
Lorsque le moteur est sur un fin de course, il ne peut plus aller dans ce sens et il ne pourra redémarrer qu'en sens inverse.